

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



# **Localização Absoluta De Robôs Móveis Em Ambientes Industriais**

**David Da Silva Lima**

Dissertação realizada no âmbito do  
Mestrado Integrado em Engenharia Electrotécnica e de Computadores  
Major Automação

Orientador: Prof. Dr. Paulo José Cerqueira Gomes da Costa

Co-orientador: Prof. Dr. António Paulo Gomes Mendes Moreira

Junho de 2010



A Dissertação intitulada

**“LOCALIZAÇÃO ABSOLUTA DE ROBÔS MÓVEIS EM AMBIENTES INDUSTRIAIS”**

foi aprovada em provas realizadas em 20/Julho/2010

o júri

  
Presidente Professor Doutor Aníbal Castilho Coimbra de Matos

Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da  
Faculdade de Engenharia da Universidade do Porto

  
Professor Doutor Andre ~~Gustavo~~ Scolari Conceição

Professor Departamento de Engenharia Elétrica da Escola Politécnica - UFBA

  
Professor Doutor Paulo José Cerqueira Gomes da Costa

Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da  
Faculdade de Engenharia da Universidade do Porto

  
Professor Doutor António Paulo Gomes Mendes Moreira

Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da  
Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

  
David da Silva Lima.

Autor - David da Silva Lima





# Resumo

A robótica móvel tem despertado grande interesse científico, revelando ser uma área em forte expansão e com perspectivas de futuro. Na indústria, os robôs móveis executam tarefas de movimentação, transporte e armazenamento de matérias-primas e produtos. Desta forma, pretende-se otimizar os fluxos de materiais e aumentar os índices de produtividade, bem como libertar operadores humanos de tarefas monótonas e repetitivas em ambientes potencialmente perigosos.

No entanto, a capacidade dos robôs móveis executarem, sem intervenção humana, as missões para as quais foram destinados está fortemente dependente de uma localização robusta e precisa em tempo real. Em ambiente industrial, a solução de localização mais comum é a do tipo filoguiado. Porém, este sistema apenas permite a execução de trajectórias fixas. Para trajectórias mais flexíveis utilizam-se, na indústria, sistemas baseados em triangulação e/ou trilateração *laser*. Contudo, é uma solução relativamente cara e que obriga à instalação de diversos reflectores delicados ao longo das trajectórias possíveis.

Neste trabalho, propõe-se implementar e testar técnicas alternativas de localização absoluta de robôs móveis em ambiente industrial. São apresentadas soluções de custo moderado, robustas e de fácil instalação e reconfiguração.

Assim, é descrito um sistema de localização assente na detecção de marcadores através de visão artificial. Adicionalmente, apresenta-se um método de localização absoluta baseado no reconhecimento de marcadores em relevo do tipo código de barras. A fusão da informação de localização é realizada por um Filtro de *Kalman* Estendido.

Um outro objectivo deste projecto é o desenvolvimento de uma solução fiável e segura para detecção de obstáculos.

Os resultados deste trabalho foram aplicados e testados num robô móvel industrial no âmbito das actividades de investigação do grupo ROBIS - Robótica e Sistemas Inteligentes do INESC Porto, LA em colaboração com a empresa DELTAMATIC - Engenharia e Automação Industrial, S. A.. Apresentam-se ainda as tecnologias utilizadas para a construção deste protótipo.



# Abstract

Mobile robotics has attracted great scientific interest, turning out to be a strong expansion and future prospect area. In industry, mobile robots perform handling, transportation and storage of raw materials and products tasks. This is expected to optimize material flows and increase productivity rates, as well as freeing human operators from monotonous and repetitive tasks in potentially dangerous environments.

However, the ability of mobile robots to execute the tasks which they were meant to perform, without human intervention, relies heavily on a robust and accurate real time location. In industrial environment, wire guidance is the most common solution. However, this system only allows the execution of fixed paths. For more flexible pathways, systems based on laser triangulation and / or trilateration, are used in industry. However, it's a relatively expensive solution and requires the installation of several delicate reflectors along possible paths.

This work proposes to implement and test absolute location alternative techniques for mobile robots in industrial environments. Moderate cost, robust and easy installation and reconfiguration solutions are presented.

Thus, a location system based on landmarks detection through artificial vision is described. Additionally, a method of absolute location based on bar code's landmarks recognition is presented. Location data fusion is performed by a Extended Kalman Filter.

Another aim of this project is the development of a reliable and safe solution for obstacle detection.

The results of this work were applied and tested in an industrial robot as part of the research activities of the ROBIS group - Robótica e Sistemas Inteligentes of INESC Porto, LA in cooperation with the firm DELTAMATIC - Engenharia e Automação Industrial, S. A.. The technologies used to build this prototype are also presented.



# Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais e irmão o apoio, formação e motivação dados ao longo da minha vida, que contribuíram de forma crucial para os meus êxitos pessoais e académicos.

Agradeço ao Prof. Dr. Paulo Costa toda a ajuda prestada ao longo da realização deste trabalho. Estou igualmente grato ao Prof. Dr. António Paulo Moreira por me propor esta dissertação e pela disponibilidade em acompanhar e auxiliar a concretização da mesma.

Aos meus colegas, agradeço a amizade e os bons momentos vividos ao longo da minha formação académica.

David da Silva Lima



# Conteúdo

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Agradecimentos</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento e Motivação . . . . .	1
1.2 Objectivos . . . . .	3
1.3 Estrutura do documento . . . . .	3
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Introdução . . . . .	5
2.2 Localização de robôs móveis em ambientes industriais . . . . .	6
2.2.1 Soluções comerciais tradicionais . . . . .	6
2.2.2 Outras soluções alternativas . . . . .	10
2.3 Soluções de detecção de obstáculos de robôs móveis industriais . . . . .	15
2.3.1 Soluções baseadas em pára-choques mecânicos . . . . .	15
2.3.2 Soluções baseadas em <i>laser</i> . . . . .	16
2.3.3 Soluções baseadas em sensores ultra-sónicos . . . . .	17
<b>3 Robô móvel industrial</b>	<b>19</b>
3.1 Apresentação Geral . . . . .	19
3.2 Plataforma de <i>Hardware</i> . . . . .	22
3.2.1 Sistema de tracção . . . . .	22
3.2.2 Plataforma de transporte . . . . .	22
3.2.3 Controladores dos motores . . . . .	22
3.2.4 Sistema de visão artificial . . . . .	25
3.2.5 Laser Range Finder . . . . .	29
3.2.6 Sistema computacional . . . . .	30
3.2.7 Sistema de alimentação . . . . .	30
3.3 Plataforma de <i>Software</i> e Comunicações . . . . .	31
3.4 Detecção de Obstáculos . . . . .	33
3.4.1 Técnicas Complementares . . . . .	34
<b>4 Localização</b>	<b>37</b>
4.1 Localização relativa - Odometria . . . . .	37
4.1.1 Modelo da Odometria . . . . .	38
4.1.2 Erros sistemáticos e calibração da odometria . . . . .	39

4.1.3	Erros não sistemáticos e modelo de erro da odometria . . . . .	40
4.2	Localização absoluta - Marcadores visuais . . . . .	43
4.3	Localização absoluta - Marcadores em relevo . . . . .	45
4.4	Fusão sensorial . . . . .	48
4.4.1	Implementação do Filtro de <i>Kalman</i> Estendido . . . . .	48
<b>5</b>	<b>Sistema de detecção de marcadores visuais</b>	<b>53</b>
5.1	Detecção da orla do marcador . . . . .	55
5.2	Detecção dos segmentos de recta do marcador . . . . .	59
5.3	Calibração da câmara . . . . .	64
5.3.1	Modelo da câmara . . . . .	64
5.3.2	Calibração dos Parâmetros de Transformação . . . . .	67
5.4	Identificação do marcador . . . . .	69
5.5	Cálculo da incerteza . . . . .	71
5.6	Identificação do código do marcador . . . . .	71
<b>6</b>	<b>Testes e resultados experimentais</b>	<b>75</b>
6.1	Testes ao sistema de detecção de marcadores visuais . . . . .	75
6.1.1	Testes de robustez . . . . .	80
6.1.2	Testes de ruído . . . . .	81
6.1.3	Velocidade máxima do robô que permite detecção . . . . .	82
6.1.4	Percurso de teste . . . . .	87
6.2	Análise ao erro de medida do <i>Laser Range Finder</i> . . . . .	90
6.3	Testes ao sistema de detecção dos marcadores em código de barras . . . . .	91
6.3.1	Teste em condições de funcionamento . . . . .	92
6.4	Teste à detecção de obstáculos . . . . .	95
<b>7</b>	<b>Conclusões</b>	<b>97</b>
7.1	Trabalho Futuro . . . . .	98
<b>A</b>	<b>Intersecção de rectas definidas por dois pontos</b>	<b>99</b>
	<b>Referências</b>	<b>101</b>



# Lista de Figuras

2.1	Método de localização baseada em filoguiado [1]	6
2.2	Princípio de funcionamento do filoguiado	6
2.3	Método de localização baseado em <i>laser</i> [1, 2]	7
2.4	Método de localização baseado no seguimento de faixas [1, 3]	8
2.5	Método de localização baseado em marcadores [1]	8
2.6	Método de localização baseado em <i>GPS</i>	9
2.7	Arquitectura típica de um sistema de localização baseado em <i>differential GPS</i>	10
2.8	Método de localização do sistema <i>Ubisense</i>	11
2.9	Abordagens à navegação de robôs móveis baseada em visão, segundo <i>DeSouza et. al.</i>	12
2.10	Robôs móveis da KIVA systems - pormenor dos marcadores de códigos de barras [4]	14
2.11	Exemplo de modelo tridimensional criado pelos robôs <i>Seegrid</i> [5]	14
2.12	Deteccção de obstáculos através de pára-choques mecânicos [6, 7]	15
2.13	Sensor <i>laser</i> para detecção de obstáculos	16
2.14	Deteccção de obstáculos utilizando sensores <i>laser</i> [6]	16
2.15	Princípio de funcionamento dos sensores de ultra-sons	17
3.1	Robô móvel industrial	19
3.2	Dimensões e estrutura básica do robô móvel	20
3.3	Arquitectura funcional dos principais sistemas do robô móvel	21
3.4	Esquema de ligações físicas <i>serial Daisy Chain</i>	21
3.5	Configuração dos motores utilizados para locomoção do robô	22
3.6	Placa controladora dos motores do robô	23
3.7	Arquitectura do conjunto controladores e sistema de tracção	23
3.8	Resposta ao degrau e estimativa dada pelo modelo aproximado de 1ª ordem	25
3.9	Câmara USB monocromática da <i>Imaging Source</i>	26
3.10	Padrão de <i>Bayer</i> [8]	26
3.11	Exemplo de conversão do formato <i>Bayer</i> para RGB [8]	26
3.12	Dimensões da cena a captar pela câmara	27
3.13	Posicionamento da iluminação artificial e da câmara	28
3.14	Varrimento efectuado por um <i>Laser Range Finder</i>	29
3.15	<i>Hokuyo URG-04LX-UG01 - Laser Range Finder</i> utilizado	30
3.16	Layout da aplicação de interface com o <i>Laser Range Finder</i>	32
3.17	Interface gráfica da aplicação de detecção dos marcadores visuais	32
3.18	Interface gráfica da aplicação de controlo do robô	33
3.19	Zonas de segurança para detecção de obstáculos	34

4.1	Robô diferencial no sistema coordenado global . . . . .	38
4.2	Forma geométrica do marcador visual escolhido . . . . .	43
4.3	Correcção da pose do robô a partir de um marcador visual . . . . .	45
4.4	Codificação do marcador em relevo . . . . .	46
4.5	Exemplo dos códigos usados . . . . .	46
4.6	Cálculo do centro e da orientação do marcador em relevo no referencial do <i>laser</i> .	47
4.7	Exemplo de um protótipo do marcador de código de barras (código 20) . . . . .	48
4.8	Resumo da operação do Filtro de <i>Kalman</i> Extendido . . . . .	52
5.1	Coordenadas e dimensões de uma <i>frame</i> . . . . .	53
5.2	Exemplo de uma <i>frame</i> captada com indicação da região de interesse . . . . .	54
5.3	Grelha usada para avaliação do efeito da distorção em barril . . . . .	55
5.4	Resultado do processo de binarização após aplicação do operador de <i>Sobel</i> sobre uma <i>frame</i> . . . . .	56
5.5	Resultado do processo de binarização após aplicação do operador de derivada de primeira ordem sobre uma <i>frame</i> . . . . .	57
5.6	Discretização da orientação do gradiente . . . . .	58
5.7	Representação binária dos pontos de orla antes e após a supressão dos não máximos locais . . . . .	59
5.8	Uma recta no plano da imagem e exemplo da transformada de <i>Hough</i> para três dos seus pontos . . . . .	60
5.9	Exemplo do resultado da transformada de <i>Hough</i> . . . . .	61
5.10	Algoritmo de pesquisa dos segmentos de recta do marcador . . . . .	62
5.11	Deteção das rectas da imagem a partir dos pontos de orla . . . . .	64
5.12	Projeção perspectiva no modelo <i>pinhole</i> [9] . . . . .	65
5.13	Sistemas coordenados especificados [10] . . . . .	65
5.14	Rotação do referencial da câmara em relação ao sistema coordenado do robô . . .	68
5.15	Algoritmo de identificação e validação do marcador . . . . .	69
5.16	Pormenor da identificação dos vértices, centro e orientação do marcador . . . . .	70
5.17	Exemplo de marcador com as zonas de código . . . . .	71
5.18	Algoritmo de identificação do código do marcador . . . . .	72
5.19	Código identificador do marcador . . . . .	73
5.20	Exemplo dos códigos usados . . . . .	74
6.1	Tempo médio de processamento do algoritmo para diferentes configurações da transformada de <i>Hough</i> , utilizando a resolução total de imagem (610 pontos transformados) . . . . .	77
6.2	Evolução do erro da detecção do marcador visual com a resolução da transformada de <i>Hough</i> . . . . .	77
6.3	Tempo médio de processamento do algoritmo para diferentes configurações da transformada de <i>Hough</i> , utilizando metade da resolução de imagem (381 pontos transformados) . . . . .	78
6.4	Exemplo de ocultação dos 3 vértices do marcador . . . . .	80
6.5	Exemplo de ocultação de grande parte da área do marcador . . . . .	80
6.6	Exemplo da rejeição de falso positivo . . . . .	81
6.7	Resultados dos testes de degradação a 75% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 0,8cm, \Delta\theta = 1grau$ ), utilizando a resolução total de imagem . . . . .	83

6.8	Resultados do testes de degradação a 100% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 0,8cm, \Delta\theta = 1grau$ ), utilizando a resolução total de imagem . . . . .	83
6.9	Resultados do testes de degradação a 75% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 1,7cm, \Delta\theta = 2grau$ ), utilizando a resolução total de imagem . . . . .	84
6.10	Resultados do testes de degradação a 100% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 1,7cm, \Delta\theta = 2grau$ ), utilizando a resolução total de imagem . . . . .	84
6.11	Resultados do testes de degradação a 75% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 0,8cm, \Delta\theta = 1grau$ ), utilizando metade da resolução de imagem . . . . .	85
6.12	Resultados do testes de degradação a 100% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 0,8cm, \Delta\theta = 1grau$ ), utilizando metade da resolução de imagem . . . . .	85
6.13	Resultados do testes de degradação a 75% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 1,7cm, \Delta\theta = 2grau$ ), utilizando metade da resolução de imagem . . . . .	86
6.14	Resultados do testes de degradação a 100% dos pixels para a resolução da transformada de <i>Hough</i> ( $\Delta\rho = 1,7cm, \Delta\theta = 2grau$ ), utilizando metade da resolução de imagem . . . . .	86
6.15	Percurso de teste executado e localização do marcador visual . . . . .	87
6.16	Estimativa da trajectória executada pelo robô, utilizando apenas a odometria . . .	88
6.17	Evolução temporal da estimativa da pose do robô, utilizando apenas odometria .	88
6.18	Estimativa da trajectória executada pelo robô, utilizando o EKF . . . . .	89
6.19	Evolução temporal da estimativa da pose do robô, utilizando o EKF . . . . .	89
6.20	Análise ao erro de medida do <i>Laser Range Finder</i> . . . . .	91
6.21	Percurso de teste dos marcadores em relevo . . . . .	93
6.22	Trajectória estimada e executada pelo robô - Percurso de teste dos marcadores em relevo . . . . .	93
6.23	Evolução temporal das posições reais e estimadas pelo robô - Percurso de teste dos marcadores em relevo . . . . .	94
6.24	Teste do sistema de detecção de obstáculos . . . . .	95
6.25	Evolução temporal da velocidade linear do robô - Teste do sistema de detecção de obstáculos . . . . .	96
6.26	Evolução temporal da posição X do robô - Teste do sistema de detecção de obstáculos	96
A.1	Ponto de intersecção de duas rectas definidas por dois pontos . . . . .	99



# Lista de Tabelas

3.1	Valores dos parâmetros do modelo dinâmico de 1ª ordem do robô . . . . .	24
3.2	Valores dos parâmetros dos controladores dos motores de tracção . . . . .	25
3.3	Formato da mensagem enviada por UDP pela aplicação de detecção de marcadores visuais . . . . .	33
3.4	Valores dos parâmetros do modelo discreto de 1ª ordem do conjunto sistema de tracção e controlador . . . . .	36
4.1	Valores dos parâmetros do modelo de odometria do robô . . . . .	40
4.2	Valores dos parâmetros do modelo de erro da odometria do robô . . . . .	42
4.3	Dimensões do marcador visual utilizado . . . . .	44
4.4	Valores dos parâmetros da matriz de covariância do ruído da medida fornecida pelos marcadores visuais . . . . .	51
4.5	Valores dos parâmetros da matriz de covariância do ruído da medida fornecida pelos marcadores em código de barras . . . . .	51
6.1	Resposta do algoritmo a 24 <i>frames</i> de teste para diferentes configurações da transformada de <i>Hough</i> , utilizando a resolução total de imagem . . . . .	76
6.2	Resposta do algoritmo a 24 <i>frames</i> de teste para diferentes configurações da transformada de <i>Hough</i> , utilizando metade da resolução de imagem . . . . .	79
6.3	Número de vezes que o algoritmo detecta o marcador visual para diferentes velocidades do robô . . . . .	82
6.4	Comparação do erro absoluto máximo ao longo do percurso, utilizando a odometria e a fusão de informação através do EKF (odometria e marcadores visuais)	90
6.5	Resultados dos testes à detecção do marcador de código de barras . . . . .	91



# Abreviaturas e Símbolos

## Lista de abreviaturas

5DPO	Grupo de Futebol Robótico da FEUP
AGV	<i>Automated Guided Vehicle</i>
AOA	<i>Angle of Arrival</i>
CCD	<i>Charge-coupled device</i>
DGPS	<i>Differential Global Positioning System</i>
DSP	<i>Digital Signal Processor</i>
EKF	<i>Extended Kalman Filter</i>
FPS	<i>Frames por segundo</i>
GPS	<i>Global Positioning System</i>
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>
PCB	<i>Printed Circuit Board</i>
RF	<i>Radio frequency</i>
RTK	<i>Real Time Kinematic</i>
SLAM	<i>Simultaneous Localization And Mapping</i>
SMD	<i>Surface Mounted Devices</i>
TDOA	<i>Time Differential Of Arrival</i>
USB	<i>Universal Serial Bus</i>
UWB	<i>Ultra-wideband</i>
WLAN	<i>Wireless Local Area Network</i>
XML	<i>eXtensible Markup Language</i>





# Capítulo 1

## Introdução

### 1.1 Enquadramento e Motivação

Até o momento, o maior sucesso da robótica tem sido alcançado na indústria da manufactura. Instalados num local fixo da linha de produção, os braços robóticos conseguem mover-se com grande velocidade e exactidão, desempenhando tarefas repetitivas como soldadura e pintura. Na indústria electrónica, estes mesmos manipuladores colocam componentes *SMD* com extrema precisão, tornando possível a produção de telemóveis e computadores portáteis. Apesar de todo o seu sucesso, os manipuladores robóticos apresentam uma grande limitação: a falta de mobilidade [11].

A robótica móvel, por seu lado, pretende aliar, no mesmo dispositivo de automação, as potencialidades de manuseamento e manipulação de um braço robótico à mobilidade de um veículo autónomo [12]. Em vez de estarem restritos a um espaço de trabalho fixo, os robôs móveis são capazes de se deslocar e cumprir as suas missões numa vasta área, sem intervenção humana. Desta forma, tornou-se possível a aplicação da robótica na realização de diversas tarefas que vão desde a vigilância e limpeza de grandes áreas até à assistência a deficientes (cadeiras de rodas autónomas), passando pela inspecção e recolha de dados dos mais variados tipos. Ao nível industrial, a principal aplicação está ligada à movimentação, transporte e armazenamento de produtos e materiais nos sistemas de produção. Os veículos guiados autonomamente, conhecidos como *AGV's*, são um exemplo de sucesso da robótica móvel no sector industrial. A sua utilização permitiu aumentar o grau de automação e integração de sistemas, bem como otimizar o fluxo de materiais nos processos de produção industrial. Desta forma, reduziram-se os tempos de transporte e os gastos energéticos, conduzindo a ganhos significativos na produtividade e eficiência. A introdução dos robôs móveis na indústria permitiu igualmente libertar operadores humanos de tarefas monótonas e desgastantes, em ambientes potencialmente perigosos (ex. indústria química e pesada).

A capacidade de se movimentar autonomamente é uma das competências mais desafiadoras exigidas a um robô móvel [11]. Para uma navegação bem sucedida é necessário que o robô retire dos seus sensores informação relevante para determinar a sua posição no meio envolvente e decidir

qual a trajectória a tomar. O problema geral da navegação pode ser resumido a três questões fundamentais: "Onde estou?", "Para onde vou?" e "Como posso chegar lá?" [13]. A localização procura responder à primeira pergunta, "Onde estou?". O seu objectivo é determinar a pose do veículo, ou seja, a posição cartesiana e a orientação do robô em relação a um referencial externo. Torna-se assim evidente que a autonomia dos robôs móveis está fortemente dependente da sua capacidade de auto-localização. Na verdade, a localização absoluta tem uma importância crucial na robótica móvel. As tarefas de planeamento e execução de trajectórias tornam-se impossíveis sem uma localização absoluta precisa e robusta em tempo real.

De modo a permitir a contínua difusão dos robôs móveis na indústria torna-se necessário ultrapassar as limitações dos sistemas actuais (apresentados no capítulo 2). Para além de uma localização robusta que permita o funcionamento autónomo dos veículos móveis durante períodos de tempo alargados, novas exigências e restrições surgem e necessitam de ser levadas em conta na concepção destes veículos. Em ambiente industrial, as soluções de localização mais comuns são do tipo filoguiado, mas possuem diversas desvantagens óbvias, sendo a principal o facto de só permitirem a execução de trajectórias fixas. Para trajectórias mais flexíveis utilizam-se, na indústria, sistemas baseados em triangulação e/ou trilateração laser. No entanto, é um sistema relativamente caro e que obriga à colocação de diversos reflectores delicados ao longo das trajectórias possíveis.

A grande competitividade dos mercados actuais, obriga as empresas a responderem rapidamente às expectativas e exigências dos clientes, assim como a desenvolver produtos mais personalizados. Aos sistemas de produção industrial é exigida a satisfação, em tempo útil, de pequenas séries, o que obriga à rápida e frequente alteração do layout<sup>1</sup> fabril. Nestes casos, os robôs móveis industriais devem ser capazes de executar trajectórias dinâmicas e rapidamente reconfiguráveis, possibilitando uma resposta adequada ao ajuste da produção. Outros factores a ter em conta são as alterações requeridas ao ambiente e os tempos de instalação do sistema, que devem ser minimizados quanto possível. Acrescente-se ainda a necessidade de desenvolver soluções de custo reduzido, que permitam a introdução destas tecnologias nas pequenas e médias empresas.

Com esta dissertação propõe-se implementar e testar técnicas alternativas de localização absoluta de robôs móveis em ambiente industrial. Pretende-se um sistema robusto e facilmente reconfigurável baseado em sensores de custo moderado. A solução deverá apresentar bastante flexibilidade, na medida em que não obrigará à modificação radical do ambiente onde o robô opera. Para isso, será desenvolvido um sistema de localização baseado em marcadores.

Os resultados deste projecto serão aplicados e testados num robô móvel industrial no âmbito do projecto MiniAGV do grupo de investigação ROBIS - Robótica e Sistemas Inteligentes do INESC Porto, LA em colaboração com a empresa DELTAMATIC - Engenharia e Automação Industrial, S. A. ([www.deltamatic.pt](http://www.deltamatic.pt)).

---

<sup>1</sup> disposição espacial dos departamentos ou postos de trabalho.

## 1.2 Objectivos

O principal objectivo deste trabalho é desenvolver e testar uma solução de localização de robôs móveis industriais, de custo moderado e baseada em marcadores artificiais. Deste modo, são objectivos desta dissertação:

- implementar um sistema de detecção de marcadores baseado em visão artificial;
- desenvolver um sistema de localização absoluta baseado em marcadores artificiais;
- apresentar um método de localização complementar, baseado no reconhecimento de marcadores em código de barras;
- implementar um sistema fiável e seguro para detecção de obstáculos;
- montar um robô móvel industrial para teste dos sistemas desenvolvidos.

## 1.3 Estrutura do documento

Este documento encontra-se dividido em 7 capítulos e pretende descrever o trabalho realizado ao longo desta dissertação.

A introdução ao documento é realizada no capítulo 1. É apresentada a problemática da localização de robôs móveis em ambientes industriais e os seus actuais desafios que motivam a realização deste trabalho. São ainda definidos os objectivos propostos para esta dissertação.

No capítulo 2 apresenta-se o estado da arte relativo às soluções mais comuns de localização e detecção de obstáculos empregues em robôs móveis em ambiente industrial. São ainda apresentados alguns trabalhos realizados com AGV's com especial enfoque na localização através de visão artificial.

O capítulo 3 apresenta os sistemas de *hardware* e as aplicações de *software* empregues na montagem do robô móvel industrial. São ainda descritas as técnicas implementadas para detecção de obstáculos durante a movimentação do veículo.

No capítulo 4 descrevem-se os diferentes métodos de localização desenvolvidos. Para além de um modelo do erro da odometria, são apresentados os algoritmos de localização absoluta do robô baseada na detecção de marcadores artificiais. O capítulo finaliza com a descrição da implementação de um Filtro de *Kalman* Estendido.

Ao capítulo 5 reserva-se a descrição do sistema de detecção de marcadores baseado em visão artificial. É apresentado o método de calibração da câmara usado e os algoritmos de processamento de imagem desenvolvidos para identificação do marcador visual e da sua posição em relação ao referencial do robô. Adicionalmente, é proposto um método de codificação unívoca do marcador.

O capítulo 6 contém os resultados obtidos durante a realização de variados testes, entre eles testes de robustez ao sistema de detecção dos marcadores.

Por fim, no capítulo 7 apresentam-se as conclusões finais e algumas sugestões de trabalho futuro.



## Capítulo 2

# Estado da Arte

### 2.1 Introdução

A localização de robôs móveis não é uma área de investigação recente. Ao longo dos últimos anos, foram propostas variadíssimas técnicas de localização, cada uma delas com os seus pontos fortes e debilidades. Esta variedade surgiu essencialmente como consequência da diversidade de tecnologias existentes e do grande leque de aplicações dadas aos robôs móveis.

No entanto, a escolha da técnica de localização a aplicar num veículo autónomo deve levar em conta, não só as características do próprio veículo e do ambiente em que vai operar, mas também o desempenho exigido durante a realização da sua missão. Factores como o tipo de trajectórias a percorrer (fixas ou dinâmicas), a precisão de navegação exigida, o custo de instalação e manutenção, a necessidade de reconfiguração frequente e a possibilidade de expansão futura do sistema condicionam a utilização das diferentes tecnologias e técnicas de localização. Sendo assim, neste capítulo será dada primazia às soluções com aplicabilidade em ambientes industriais, uma vez que é reconhecidamente impossível mencionar todos os métodos de localização de robôs móveis já estudados e propostos.

Para além de um vasto trabalho de investigação, existem actualmente no mercado diferentes soluções de localização de robôs móveis para a indústria. Algumas delas apresentam grandes níveis de fiabilidade, devido aos vários anos de maturação. Em guisa de exemplo, o primeiro AGV da história foi criado em 1950, por *Barrett Electronics*. Consistia num veículo de reboque que seguia um fio eléctrico embutido no pavimento [3]. Esta mesma técnica ainda hoje é aplicada na indústria.

Na secção 2.2 são apresentadas as soluções comerciais de localização de robôs móveis actualmente empregues em ambientes industriais, bem como algumas técnicas alternativas e avanços recentes. Os sistemas utilizados para detectar obstáculos em robôs móveis industriais são referidos na secção 2.3.

## 2.2 Localização de robôs móveis em ambientes industriais

No tópico 2.2.1 são sucintamente apresentadas as principais soluções comerciais oferecidas pelos fabricantes de robôs móveis para a indústria. Alguns desenvolvimentos recentes e técnicas de localização alternativas com possível aplicabilidade em ambientes industriais são referidas na secção 2.2.2.

### 2.2.1 Soluções comerciais tradicionais

Após consulta dos fabricantes de robôs móveis para a indústria verifica-se que as soluções de localização utilizadas são bastante semelhantes entre si. Nas secções 2.2.1.1 e 2.2.1.3 são apresentadas as técnicas empregues para seguimento de caminhos fixos, ao passo que nos pontos 2.2.1.2 e 2.2.1.4 se descrevem os métodos que permitem a execução de trajectórias dinâmicas. Uma técnica amplamente utilizada para localização em ambientes exteriores é mencionada em 2.2.1.5.

#### 2.2.1.1 Soluções baseadas em filoguiado

O filoguiado, ilustrado na figura 2.1, é uma das soluções mais difundidas na indústria e a forma mais simples de navegação. Nestas soluções é colocado um sensor por debaixo do veículo para detectar um sinal RF proveniente de um fio eléctrico previamente embutido no pavimento (a cerca de 2 a 3 centímetros da superfície do solo [14]), conforme esboçado na figura 2.2.

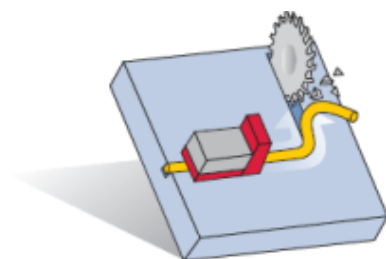


Figura 2.1: Método de localização baseada em filoguiado [1]

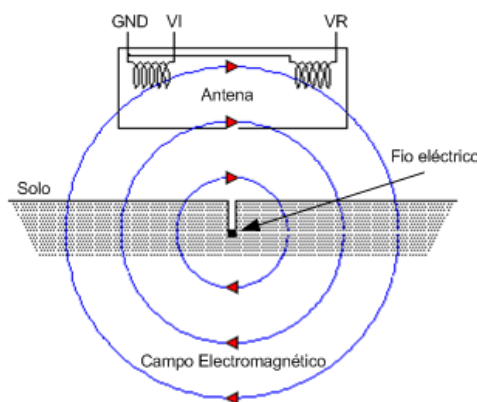


Figura 2.2: Princípio de funcionamento do filoguiado

A corrente alternada que atravessa o fio cria em torno deste um campo electromagnético. Este campo induz uma tensão aos terminais da bobina que é proporcional à sua intensidade. O sensor instalado no veículo possui duas bobinas colocadas nos dois lados do fio condutor embutido no solo. Assim, é possível obter um sinal de controlo a partir da diferença entre as tensões aos terminais de cada bobina. Quando o sensor está posicionado sobre o condutor esta diferença é nula. Assim que o veículo se desvia do caminho, a tensão aumenta numa bobina e diminui na

bobina oposta, resultando num sinal diferencial utilizado para ajustar a posição do veículo de modo a mantê-lo no caminho pré-definido [15].

Este método tem sido largamente adoptado devido à sua simplicidade e robustez [12]. No entanto, como as únicas trajectórias possíveis têm de ser previamente definidas pela introdução de um fio eléctrico no solo, qualquer modificação ao deslocamento do veículo apresenta tempos de interrupção de serviço e custos elevados. Para além disso, esta técnica é bastante sensível mesmo a pequenos desníveis do solo e obriga os veículos a movimentarem-se a velocidades relativamente baixas de modo a evitar a perda do contacto com o fio que serve de guia [16]. Apesar das suas desvantagens, estas soluções são ainda bastante usadas em aplicações que requerem um nível elevado de precisão no caminho a percorrer, como por exemplo a viagem de ida e volta entre duas estações de trabalho numa zona congestionada.

### 2.2.1.2 Soluções baseadas em triangulação e/ou trilateração *laser*

Para trajectórias flexíveis utiliza-se, na indústria, sistemas baseados em triangulação e/ou trilateração por reflectores de laser (ver figura 2.3). Os painéis reflectores são instalados em colunas, paredes ou outros locais a alguns metros acima do solo, de modo a evitar a sua obstrução. No veículo é colocado um *laser scanner* que realiza constantemente um varrimento rotativo [12]. Através dos feixes de luz reflectidos, o sistema *laser* mede a distância e/ou orientação de cada reflector em relação ao robô. Como as coordenadas dos painéis reflectores são conhecidas de antemão, é possível calcular a localização absoluta do veículo recorrendo a técnicas de triangulação e/ou trilateração.

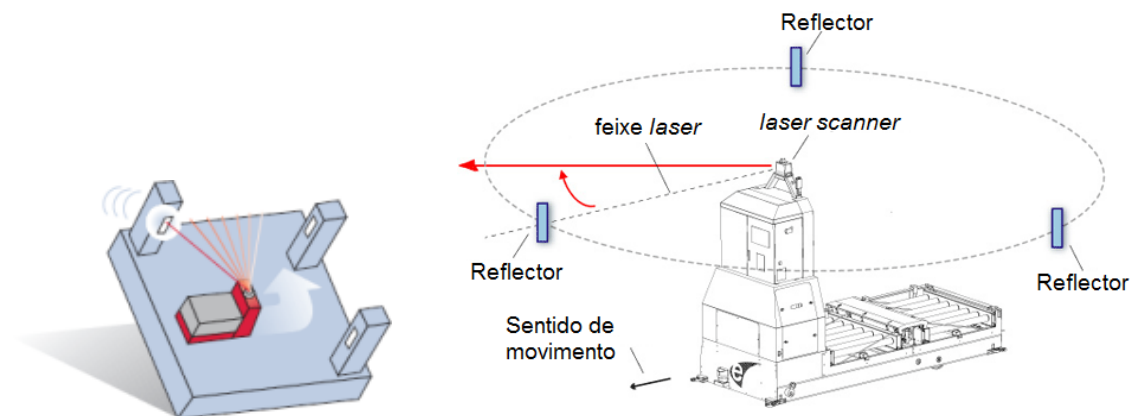


Figura 2.3: Método de localização baseado em *laser* [1, 2]

As principais vantagens destes sistemas são a sua precisão (na ordem dos milímetros), flexibilidade e velocidade de movimentação do veículo [16]. No entanto, é uma solução com custo inicial relativamente alto e que obriga à colocação de vários painéis reflectores ao longo das trajectórias possíveis. Para obtenção da localização do robô é necessário que, em cada instante de tempo, pelo menos três reflectores sejam detectados, sendo aconselhável a visibilidade de pelo menos cinco painéis [17].

### 2.2.1.3 Soluções baseadas em faixas

Nos sistemas baseados em faixas, os veículos seguem percursos definidos por fitas magnéticas, ou coloridas, colocadas sobre o piso [18]. Uma vez que não é necessário efectuar cortes no pavimento, o custo de instalação com este método é inferior ao filoguiado. Para além disso, é também uma solução mais flexível, pois permite alterar os percursos em apenas algumas horas, bastando para isso a remoção e recolocação das fitas [19]. No entanto, com esta abordagem, a localização do robô fica dependente da perfeita identificação das fitas, pelo que pode ser inviável a sua utilização em ambientes movimentados, nos quais as faixas podem ficar danificadas ou sujas.

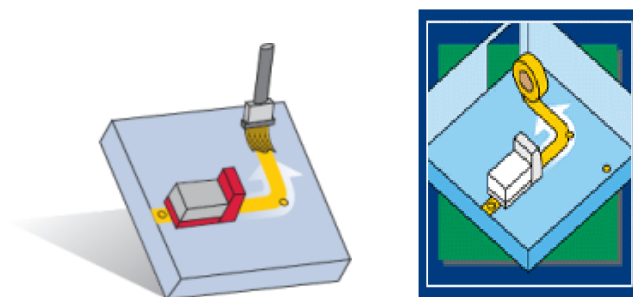


Figura 2.4: Método de localização baseado no seguimento de faixas [1, 3]

### 2.2.1.4 Soluções baseadas em marcadores e sensores inerciais

Outra possibilidade para localização de robôs móveis é a utilização conjunta de marcadores e sensores inerciais. Neste caso, o robô é equipado com um giroscópio que detecta pequenas variações de direcção, permitindo manter uma boa estimativa da orientação do veículo. No pavimento são embutidos marcadores, habitualmente pequenos ímanes ou *tags* RF passivas (ver figura 2.5). Estes dispositivos são colocados em coordenadas conhecidas e servem como pontos de referência para o robô corrigir os erros de localização acumulados no percurso entre marcadores. Normalmente a distância entre marcadores é de cerca de 8 metros [18] [14].

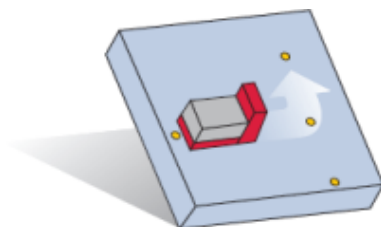


Figura 2.5: Método de localização baseado em marcadores [1]

Comparativamente aos sistemas filoguiados, esta solução permite uma instalação e reconfiguração mais rápida do *layout*, no entanto o veículo necessita de um equipamento mais caro e normalmente a precisão é inferior. Esta solução é também menos precisa do que os sistemas *laser*, mas utiliza equipamentos no próprio veículo mais simples e baratos. O custo de



instalação de pequenos ímanes no pavimento pode também ser inferior ao da colocação dos reflectores *laser* [17].

### 2.2.1.5 Soluções baseadas em GPS

A localização através de sistemas de posicionamento global, ou *GPS*, como são mais conhecidos, foi inicialmente desenvolvida em exclusivo para uso militar, mas que agora apresenta-se gratuitamente disponível para aplicações civis. Na actualidade, este método de localização encontra-se muito difundido em automóveis com sistema de navegação por mapas e é a técnica mais popular na localização de robôs móveis em ambientes exteriores.

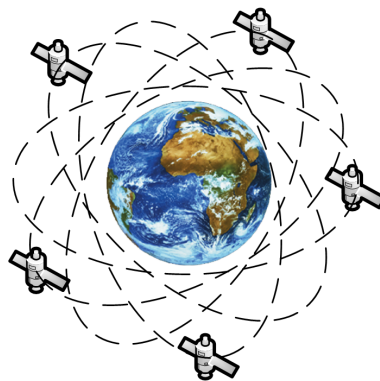


Figura 2.6: Método de localização baseado em GPS

Nesta solução, um conjunto de satélites enviam continuamente, via rádio, a sua posição global em relação ao referencial terrestre e o instante de envio da mensagem (ver figura 2.6). Os vários satélites sincronizam as suas transmissões de forma a que os seus sinais sejam enviados ao mesmo tempo. Quando um receptor *GPS* lê a transmissão de dois ou mais satélites, a diferença dos instantes de recepção das mensagens permite ao dispositivo receptor determinar a sua distância relativa a cada satélite. Combinando a informação relativa ao tempo de chegada e localização instantânea de pelo menos quatro satélites, o receptor é capaz de inferir a sua localização absoluta. Actualmente, a precisão destes sistemas, para uso civil, é de cerca de 10 metros [11].

Entretanto, foram desenvolvidas outras tecnologias que complementam o sistema *GPS* e reduzem consideravelmente os erros de estimativa da posição. Um exemplo é o *differential GPS*, ou *DGPS*, que utiliza um segundo receptor *GPS* fixo e instalado numa posição exacta conhecida (ver figura 2.7). Esta estação local recebe os sinais dos satélites *GPS* e calcula a diferença entre a estimativa de posição dada pelo método de triangulação e a sua posição exacta, que é conhecida *a priori*. Este erro é enviado aos robôs móveis que podem assim obter um cálculo mais preciso da sua localização. Com esta técnica é possível atingir uma resolução inferior a um metro, mas é necessário que o robô móvel permaneça dentro de um perímetro de alguns quilómetros ao receptor fixo. Outra técnica mais complexa, designada *RTK*, utiliza a fase dos sinais de *carrier* de cada transmissão dos satélites recebida. Com esta tecnologia é possível alcançar uma precisão na ordem de um centímetro [11].

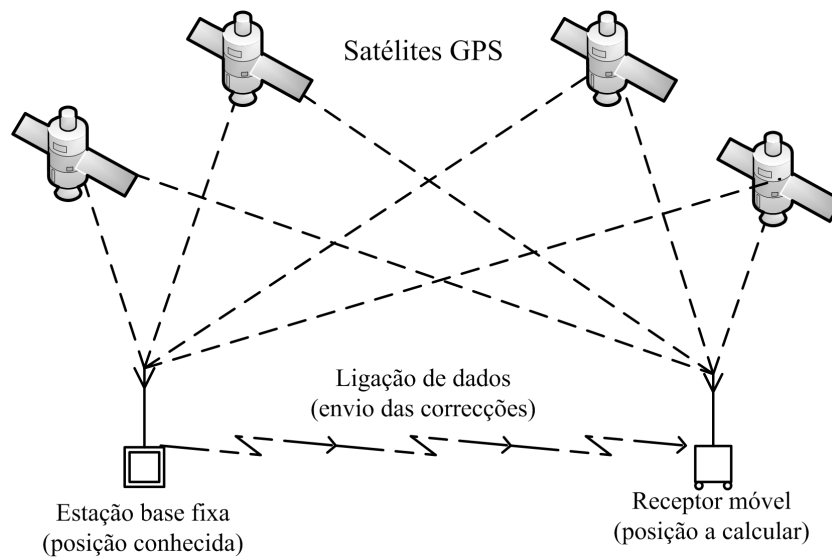


Figura 2.7: Arquitectura típica de um sistema de localização baseado em *differential GPS*

O facto de o receptor *GPS* ter de captar o sinal de pelo menos 4 satélites é uma limitação significativa. Estes sinais são de baixa potência, pelo que a sua recepção requer boa visibilidade entre o robô móvel e os satélites. Como em ambientes urbanos os edifícios podem ocultar os satélites, esta solução é habitualmente aplicada apenas quando os robôs se movimentam em ambientes exteriores amplos e é normalmente utilizada em veículos de grande porte quando não é exigida uma elevada precisão. Apesar dos sistemas complementares ao *GPS*, apresentados anteriormente, alcançarem melhor precisão, têm as desvantagens adicionais de necessitarem de estações base fixas e apresentam custos bastante mais elevados [20].

## 2.2.2 Outras soluções alternativas

Para além das soluções comerciais tradicionais, referidas em 2.2.1, outras técnicas alternativas de localização de robôs móveis encontram-se já em estudo e começam a surgir no mercado. Algumas delas aplicam-se na indústria e são referidas nas secções seguintes.

### 2.2.2.1 Soluções baseadas em tecnologias RF

A utilização de tecnologias RF na localização de veículos móveis tem sido alvo de estudo recente [21]. Alguns trabalhos sugerem a utilização de tecnologias como UWB [22], WLAN [23], ZigBee [24] e RFID. Porém, de um modo geral, a precisão destes sistemas é ainda baixa quando comparada com as soluções tradicionais [21].

Na maioria destas soluções o cálculo da localização do robô é realizado com base na informação da distância entre este e os dispositivos com que interage. Um método muito utilizado na medição desta distância passa por se saber a força do sinal recebido. Porém, em ambientes interiores, a relação entre a distância percorrida e a atenuação do sinal não é linear. Além disso, esta medida apresenta ruído não gaussiano resultante de efeitos como reflexões,

tráfego de rede, presença de pessoas ou condições atmosféricas. Estes factores tornam difícil quantificar de forma precisa a distância da fonte emissora [25]. Com o objectivo de melhorar a precisão da localização, outras soluções baseiam-se em técnicas alternativas, como por exemplo o tempo de voo do sinal, mas apresentam igualmente algumas limitações.

De referir ainda que em robôs que já fazem uso de comunicações por rádio frequência, a utilização desses sinais para localização apresenta-se bastante mais atractiva. Além de reduzir os custos com equipamentos, possibilita ainda que equipas de robôs colaborem no cálculo da localização ao mesmo tempo que trocam dados.

A nível comercial existem já soluções que fazem uso de tecnologia RF. Neste âmbito, a empresa *Ubisense* [26] apresenta um sistema de localização em tempo real baseado em tecnologia UWB. Este sistema permite a localização de dispositivos a três dimensões com uma precisão de 15 centímetros com um nível de confiança de 95%. No dispositivo a monitorizar é colocada uma *tag* activa que emite um impulso de sinais UWB, de frequência compreendida entre 6 GHz e 8,5 GHz. Uma rede de sensores é instalada no local onde se pretende localizar os dispositivos, de modo a receberem o sinal emitido pelas *tag*'s. Habitualmente, esta rede é composta por quatro sensores, sendo que um deles assume o papel de mestre e os restantes de escravos. A posição absoluta do dispositivo é calculada, no sensor mestre, através da combinação dos ângulos de recepção (AOA, *Angle of Arrival*) e da diferença entre tempos de chegada (TDOA, *Time Differential Of Arrival*) do impulso de sinal a cada sensor (ver figura 2.8). As medidas podem ser actualizadas no máximo a cada 109 milissegundos.

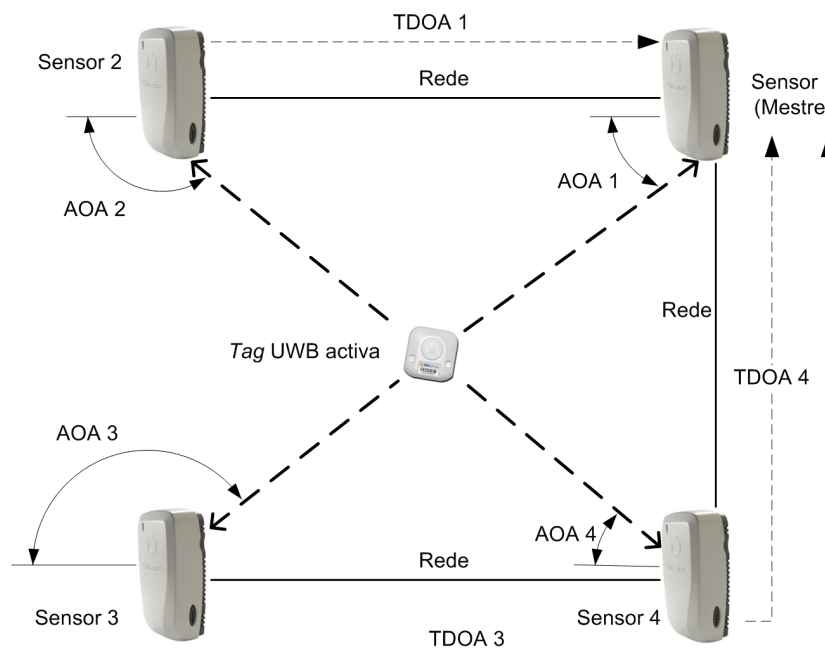


Figura 2.8: Método de localização do sistema *Ubisense*

Com uma rede de quatro sensores é possível localizar uma *tag* numa área máxima de 40x40 metros. A altura a que pode ser localizado um dispositivo é de cerca de três a cinco metros,

podendo chegar no máximo a 15 metros. No entanto, este sistema é altamente modular, permitindo a expansão da zona de monitorização através da adição de mais sensores à rede. Outra vantagem desta solução é a existência de mecanismos internos para detectar grande parte das reflexões de sinal recebidas e assim descartar dados que poderiam introduzir erros nas medidas.

Por outro lado, para que um robô móvel tome conhecimento da sua localização através deste sistema é necessário o envio dos dados através de uma rede de comunicação adicional. Além disso, o UWB não atravessa superfícies metálicas e com elevada percentagem de água, pelo que estas estruturas podem constituir um obstáculo à propagação do sinal.

### 2.2.2.2 Soluções baseadas em visão artificial

Neste tipo de sistemas, a localização do veículo é obtida através do processamento de imagens capturadas em tempo real por uma ou mais câmaras montadas no robô móvel. Este tipo de abordagem à localização tem sido alvo de um intenso trabalho de investigação, levando ao aparecimento de um vasto leque de técnicas.

*DeSouza et. al.* [27] apresenta vários trabalhos realizados na área da navegação de robôs móveis com recurso a visão artificial. Para ambientes interiores, são identificadas três principais abordagens, conforme esquematizado na figura 2.9.

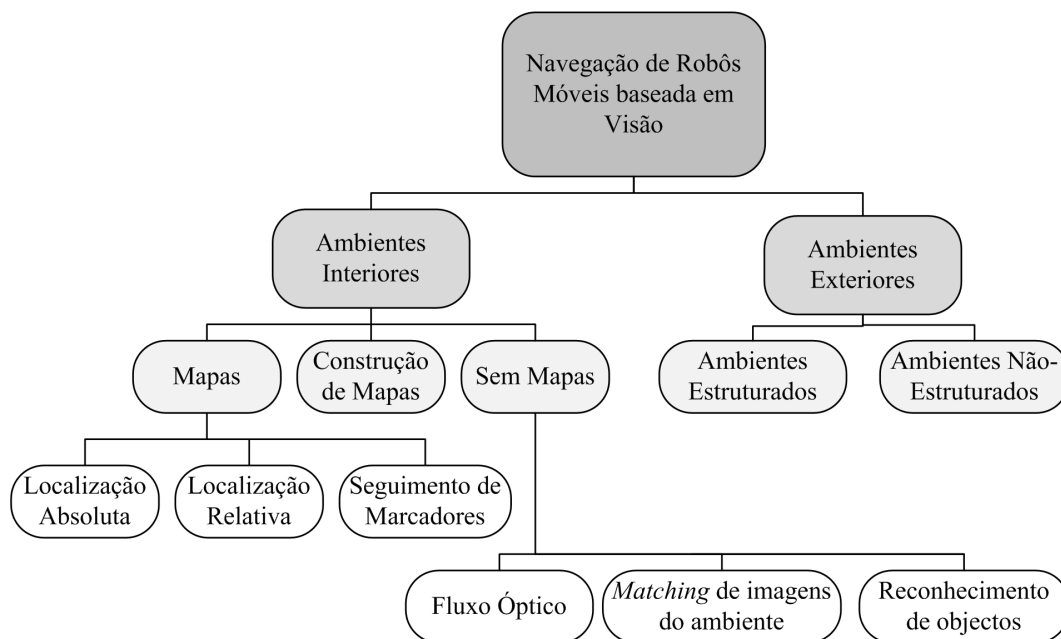


Figura 2.9: Abordagens à navegação de robôs móveis baseada em visão, segundo *DeSouza et. al.*

A primeira consiste em dotar o robô de um modelo, ou mapa, do ambiente onde vai operar. A ideia central neste tipo de navegação é prover, directa ou indirectamente, o robô de uma sequência de pontos de referência que se espera serem encontrados durante a execução das trajectórias. Desta forma, o sistema de visão fica encarregue de procurar e identificar esses pontos de referência, ou marcadores. Uma vez identificados, o robô pode comparar a observação com a informação dos

marcadores guardada numa base de dados e assim estimar a sua posição actual. Outra abordagem propõe automatizar a construção dos modelos. Neste tipo de navegação, baseada em construção de mapas, o robô é responsável por explorar o ambiente e construir uma representação interna do mesmo. Por fim, existem sistemas que não utilizam mapas durante a navegação. Os movimentos do robôs são determinados pela simples observação dos elementos envolventes, como paredes ou portas.

Para aplicações industriais, existem igualmente alguns trabalhos de investigação que propõem métodos de localização baseados, em parte ou na totalidade, em visão artificial.

*Jun et. al.* [28] apresenta um sistema de navegação de um AGV cuja trajectória é definida por uma linha pintada no pavimento. As imagens capturadas por uma câmara monocular são processadas por um *DSP*, que calcula a distância e a orientação do veículo em relação à linha. Os pontos de paragem são assinalados através de algarismos arábicos colocados ao lado da guia e igualmente reconhecidos por visão artificial.

*P. Carvalho et. al.* [15] desenvolveram um protótipo de um AGV que utiliza uma câmara colocada lateralmente no veículo para identificar marcadores visuais do tipo código de barras. O algoritmo de processamento de imagem permite calcular a distância e orientação do robô em relação ao marcador. Como a posição no referencial global do marcador visual é conhecida, a localização absoluta do veículo fica determinada pela detecção de apenas um código. No entanto, este sistema só funciona quando o AGV passa relativamente perto dos marcadores. Assim, surge a necessidade de complementar esta técnica de localização com outro tipo de informação sensorial, como a odometria.

*R. Cucchiara et. al.* [29] sugere a utilização de visão estereoscópica. A localização absoluta do AGV é obtida pela junção de duas técnicas. Por um lado, são usadas linhas paralelas ou perpendiculares aos eixos coordenados, pintadas no pavimento, para corrigir a orientação do robô. Por outro lado, o sistema utiliza as imagens capturadas pela câmara estereoscópica para calcular a distância a marcadores artificiais com características de auto-similaridade colocados nas paredes ou no pavimento. Esta informação permite a correcção da estimativa da posição cartesiana do veículo.

Outro trabalho com um AGV propõe o uso de visão para seguimento dos limites da faixa rodoviária num sistema de navegação em ambiente exterior [30]. Em [31] é proposto um sistema de posicionamento de um AGV no local de estacionamento com recurso à detecção de uma figura visual colorida.

Apesar de todo o trabalho de investigação realizado, a utilização de visão artificial nas soluções comerciais de robôs móveis industriais, tem sido escassa. No entanto, é uma solução com um futuro muito promissor e que tem despertado um grande interesse por parte da comunidade científica. Isto acontece porque as câmaras possibilitam a recolha de uma grande quantidade de informação, sem interferir no meio envolvente, e apresentam actualmente uma relação qualidade/preço muito atractiva. Aliás, começam a surgir no mercado os primeiros robôs móveis para ambientes industriais com localização baseada em visão.

Por exemplo, o sistema de transporte e manuseamento de materiais da empresa *KIVA Systems* [4] é composto por robôs móveis equipados com uma câmara que detecta códigos de barras instalados no pavimento em intervalos regulares, conforme se destaca na figura 2.10 [32]. A navegação efectua-se sobre uma espécie de grelha virtual bidimensional definida pelos marcadores do tipo código de barras.



Figura 2.10: Robôs móveis da KIVA systems - pormenor dos marcadores de códigos de barras [4]

Entretanto, a empresa *Seegrid* [7] anuncia ter desenvolvido o primeiro e único robô móvel industrial que se movimenta através de visão artificial. No robô são instaladas 8 câmaras, fornecendo visão estereoscópica ao veículo em 4 direcções. Os caminhos a percorrer pelo veículo são previamente indicados por demonstração. Nesta fase, à medida que o robô percorre as trajectórias pretendidas, são captadas imagens do meio envolvente e criados modelos tridimensionais semelhantes ao da figura 2.11. Após programação, a navegação do robô baseia-se na exploração destes modelos do ambiente, utilizando métodos probabilísticos, como o filtro de Bayes [5].



Figura 2.11: Exemplo de modelo tridimensional criado pelos robôs Seegrid [5]



Diferentemente de um sistema *SLAM*, de mapeamento e localização simultâneo, o robô apenas é capaz de repetir os percursos que tenham sido previamente memorizados. No entanto, esta solução exemplifica a flexibilidade e o potencial da utilização da visão artificial em soluções de localização de robôs móveis.

## 2.3 Soluções de detecção de obstáculos de robôs móveis industriais

Um aspecto de importância crucial a ter em conta na concepção de robôs móveis é a segurança. Em ambiente industrial, os robôs móveis muitas vezes operam em ambientes movimentados, partilhando o local de trabalho com operadores humanos e outros sistemas autónomos. Assim, para que a navegação destes veículos se processe em segurança, garantindo a integridade do meio envolvente, são necessárias soluções confiáveis que permitam a detecção atempada de potenciais obstáculos.

Os sistemas comerciais de detecção de obstáculos actualmente empregues em robôs industriais são apresentados nas secções seguintes. Alguns trabalhos de investigação realizados com AGV's sugerem a utilização de visão artificial, fazendo uso de, por exemplo, câmaras que fornecem informação de distância (baseada no princípio do tempo de voo) a três dimensões [33] ou visão estereoscópica [29]. No entanto, estas soluções ainda não têm grande expressividade no mercado.

### 2.3.1 Soluções baseadas em pára-choques mecânicos

Os pára-choques mecânicos, como ilustrados na figura 2.12, são tipicamente feitos de material flexível, como por exemplo plástico policarbonato, e ao entrarem em contacto com algum obstáculo deformam-se e accionam um interruptor que faz parar o veículo. Tipicamente o robô só continua a sua operação após reinício manual do sistema.

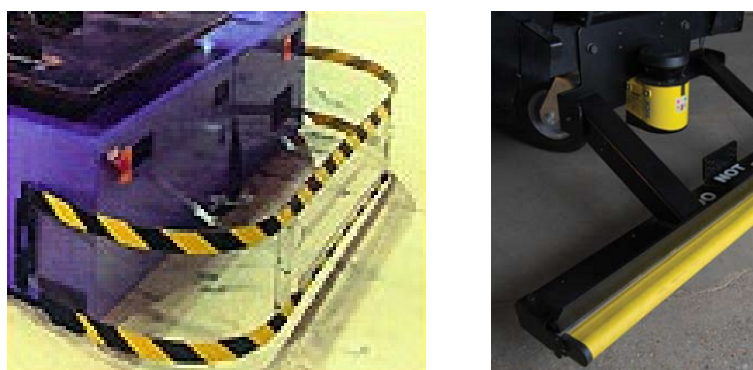


Figura 2.12: Detecção de obstáculos através de pára-choques mecânicos [6, 7]

Em locais onde o espaço disponível e a manobrabilidade o permitem, esta abordagem é uma solução de detecção de obstáculos viável e de custo acessível [6]. A sua zona de detecção é fixa e permanece constante durante o funcionamento do veículo [34]. Após detecção da colisão, o veículo leva ainda algum tempo a parar, tempo esse que depende da velocidade do mesmo.

Assim, de modo a garantir a segurança de pessoas e bens, o robô deve circular a velocidades suficientemente baixas, o que pode representar uma limitação com graves implicações ao nível da produtividade.

### 2.3.2 Soluções baseadas em *laser*

Esta solução utiliza sensores *laser*, semelhantes ao da figura 2.13, para detectar a presença de obstáculos dentro da sua zona de alcance e pode ser utilizada em simultâneo com os pára-choques mecânicos. Apesar do custo relativamente elevado, esta técnica é a forma mais usual de detecção de obstáculos, caracterizada por grande flexibilidade e ausência de necessidade de manutenção [34].



Figura 2.13: Sensor *laser* para detecção de obstáculos

Com este sistema é possível definir diferentes zonas de detecção de obstáculos, conforme ilustrado na figura 2.14, podendo estas ser alteradas durante a operação do robô. A cada zona de segurança é possível atribuir uma diferente actuação por parte do veículo em caso de presença de obstáculos. Assim, para zonas consideradas de baixo perigo o veículo pode reduzir a sua velocidade ou iniciar o contorno do obstáculo, ao passo que a presença de obstáculos em zonas de segurança máxima faz parar de imediato o robô. Este método é capaz de detectar obstáculos até 15 metros e sem a existência de contacto físico, permitindo que o veículo se movimente a velocidades mais elevadas do que as alcançadas pelos sistemas que utilizam pára-choques mecânicos.

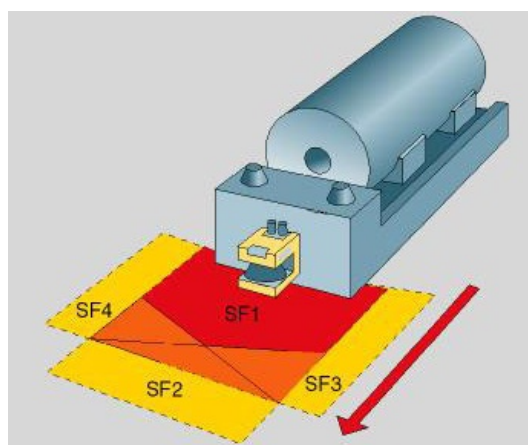


Figura 2.14: Detecção de obstáculos utilizando sensores *laser* [6]



### 2.3.3 Soluções baseadas em sensores ultra-sônicos

Nas soluções baseadas em sensores ultra-sônicos utiliza-se o tempo de propagação de uma onda sonora para medir a distância a objectos. No robô são instalados pelo menos um emissor e um receptor de ultra-sons. Um sinal com uma determinada frequência, normalmente na gama dos 40 KHz, é gerado para ser transmitido pelo emissor de ultra-sons. Na presença de obstáculos esta onda é reflectida e detectada pelo receptor de ultra-sons, conforme ilustrado na figura 2.15. Um controlador mantém o registo do tempo decorrido desde o envio do sinal pelo emissor até à detecção do eco pelo receptor e, conhecendo a velocidade do som, calcula a distância a que o obstáculo se encontra.

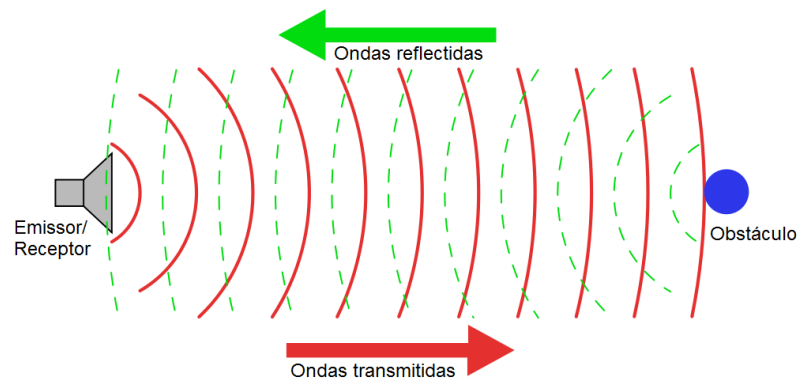


Figura 2.15: Princípio de funcionamento dos sensores de ultra-sons

Apesar de estar a ser largamente utilizada na indústria automóvel, esta solução ainda é pouco comum em veículos móveis industriais. No entanto, esta tecnologia é já usada por alguns sistemas comerciais para a detecção de obstáculos até 4 metros do veículo [35, 36]. Esta abordagem é por vezes sugerida pelos fabricantes de AGV's para complementar os pára-choques mecânicos e como alternativa às soluções baseadas em *laser* [37].



## Capítulo 3

# Robô móvel industrial

Neste capítulo será apresentado o robô móvel concebido para aplicação e teste dos sistemas de localização e detecção de obstáculos desenvolvidos. Serão brevemente referidas as suas características gerais, as plataformas de *hardware* e *software* empregues para controlo do veículo, bem como outros sistemas concebidos durante a realização deste projecto.

### 3.1 Apresentação Geral

O robô móvel industrial concebido para teste e validação dos algoritmos desenvolvidos neste trabalho está representado na figura 3.1. Este veículo foi projectado tendo em mente o transporte de materiais na indústria, pelo que as suas dimensões correspondem às de uma paleta europeia, segundo a norma *ISO 6780*, isto é, 120 x 80 centímetros.

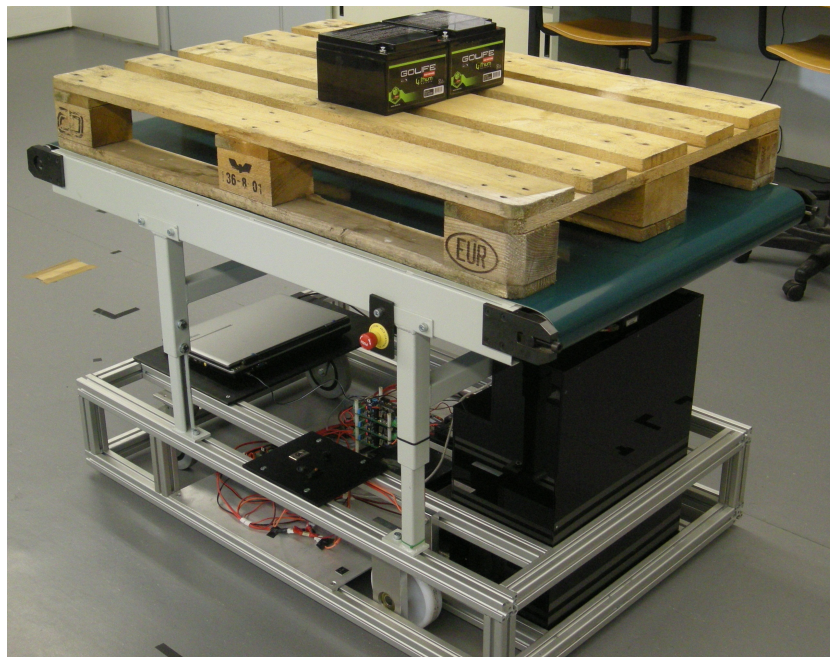


Figura 3.1: Robô móvel industrial

Os objectos a transportar podem ser colocados directamente sobre a plataforma superior do veículo ou fazendo uso do tapete rolante que a constitui. Além de permitir o deslocamento da carga em ambos os sentidos, esta plataforma pode ainda ser regulada em altura (ver figura 3.2(a)), pelo que demonstra ser bastante versátil quando em interacção com, por exemplo, outros robôs móveis semelhantes ou sistemas de transporte baseados em tapetes rolantes.

O robô movimenta-se com base num sistema de tracção diferencial dianteira e possui na retaguarda duas rodas livres do tipo castor que permitem a sustentação do veículo (ver figura 3.2(b)). Esta configuração torna o veículo bastante estável e, como possui apenas duas rodas motrizes, é de fácil controlo. Ao mesmo tempo, apresenta um bom nível de manobrabilidade, possibilitando inclusivamente que o robô rode sobre si. Na secção 3.2.1 é feita uma breve referência ao *hardware* utilizado para locomoção do veículo.

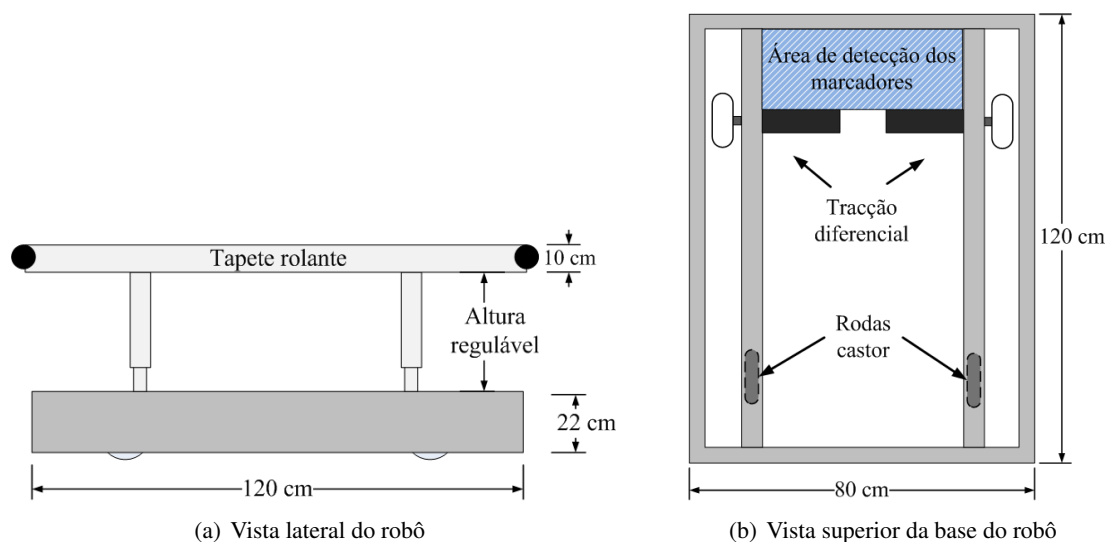


Figura 3.2: Dimensões e estrutura básica do robô móvel

Para que o robô móvel seja dotado de autonomia e cumpra as tarefas que lhe forem confiadas, foi necessário instalar, desenvolver e interligar vários equipamentos electrónicos e aplicações de *software*. A arquitectura funcional dos principais sistemas que o constituem encontra-se representada na figura 3.3.

O controlo do sistema de tracção e da plataforma de transporte de objectos é realizado por três controladores que, para além da leitura de sensores e actuação nos respectivos motores, executam algumas tarefas de processamento (ver secção 3.2.3). Estes dispositivos interagem com a aplicação central de controlo através de uma ligação *serial Daisy-Chain*. Esta conexão adapta a ligação série RS-232 a uma topologia em anel, permitindo interligar mais de dois dispositivos através da mesma ligação física (ver figura 3.4). O protocolo de comunicações utilizado, designado *Daisy Bin*, implementa uma arquitectura do tipo *Master-Slave*, sendo o dispositivo *Master* responsável pelo início da conversação e o único habilitado a efectuar pedidos. Para além de um dispositivo *Master*, uma rede *serial Daisy Chain* permite interligar um número variável de dispositivos *slaves*. No caso deste projecto, os controladores dos três motores desempenham o papel de Slaves, sendo

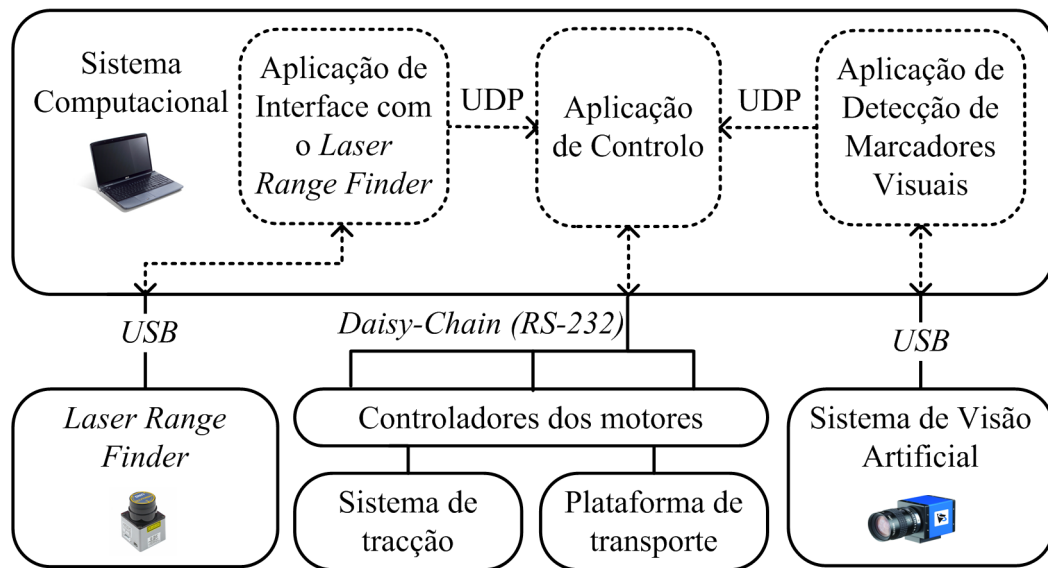


Figura 3.3: Arquitectura funcional dos principais sistemas do robô móvel

o papel de *Master* assumido pela aplicação de controlo do veículo. Esta solução de comunicação tem sido amplamente utilizada pela equipa 5DPO, do grupo de investigação ROBIS do INESC Porto, nos seus projectos de robótica móvel. Mais informações relativas à ligação *serial Daisy Chain* e ao protocolo *Daisy Bin* podem ser consultadas na referência [3].

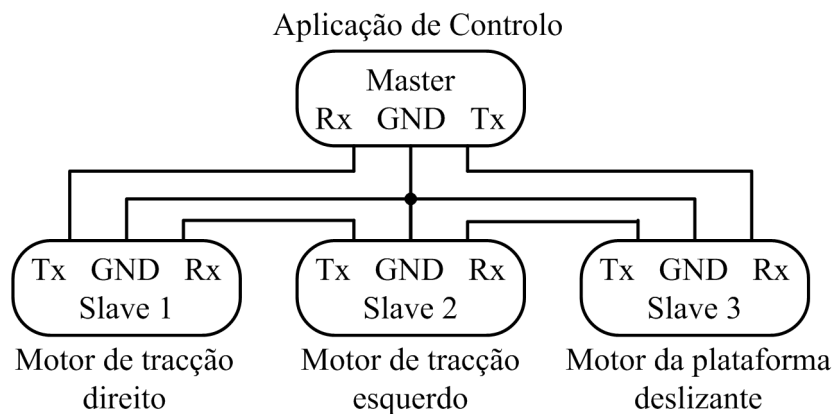


Figura 3.4: Esquema de ligações físicas *serial Daisy Chain*

O sistema de visão artificial, apresentado na secção 3.2.4, é responsável por captar imagens de uma zona livre, situada na parte dianteira do veículo, onde se pretende detectar marcadores visuais que auxiliam a localização do robô (ver figura 3.2(b)). Na zona frontal, foi instalado um *Laser Range Finder* (ver secção 3.2.5) que permite adquirir dados usados para detecção de obstáculos e localização absoluta. As aplicações de *software* criadas para controlo, localização e navegação do robô são executadas num sistema computacional, apresentado na secção 3.2.6.

## 3.2 Plataforma de *Hardware*

### 3.2.1 Sistema de tracção

Conforme referido anteriormente, o robô movimenta-se com base num sistema de tracção diferencial. Para isso recorreu-se a dois motores DC, da *Maxon* [38], com uma tensão de alimentação nominal de 24 V, proporcionando 150 W de potência cada. O acoplamento às rodas é feito através de uma caixa redutora, cuja relação de transformação é de 43:1. Desta forma, é possível transmitir às rodas um binário superior à custa da redução de velocidade.

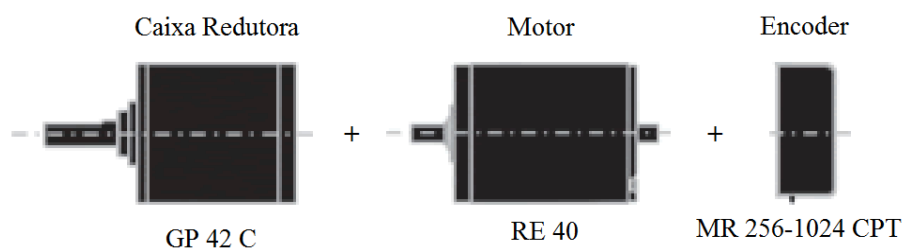


Figura 3.5: Configuração dos motores utilizados para locomoção do robô

De modo a quantificar a distância percorrida e a velocidade instantânea de cada roda foram acoplados *encoders* ao veio de cada um dos motores. A sua resolução é de 256 impulsos por cada revolução do veio, isto é, rotação de 360°. No entanto, como estes foram acoplados directamente ao veio do motor e não após a caixa redutora, cada volta da roda gerará 43 vezes mais impulsos. Este aumento do número de impulsos por rotação permite aumentar a resolução dos cálculos de velocidade e distância percorrida por cada roda.

### 3.2.2 Plataforma de transporte

Os objectos a movimentar pelo robô são transportados sobre a sua plataforma superior. Conforme mencionado anteriormente, a altura desta plataforma em relação à base do veículo pode ser ajustada mecanicamente, ao passo que o tapete rolante que a constitui permite o deslocamento da carga em ambos os sentidos durante o normal funcionamento do robô. A movimentação do tapete é efectuada por um motor DC de ímanes permanentes, alimentado a 24 V e de potência nominal de 90 W. O controlo deste motor é realizado com a mesma placa controladora dos motores de tracção.

### 3.2.3 Controladores dos motores

Para controlar os motores de locomoção do robô e movimentação da plataforma de transporte recorreu-se a placas controladoras desenvolvidas pelo grupo de investigação ROBIS do INESC Porto. Estas placas, semelhantes à apresentada na figura 3.6, têm sido amplamente usadas com sucesso nos seus projectos de robótica móvel, incluindo na sua equipa de futebol robótico 5DPO (*Small and Middle Size League*), daí a sua utilização neste projecto.

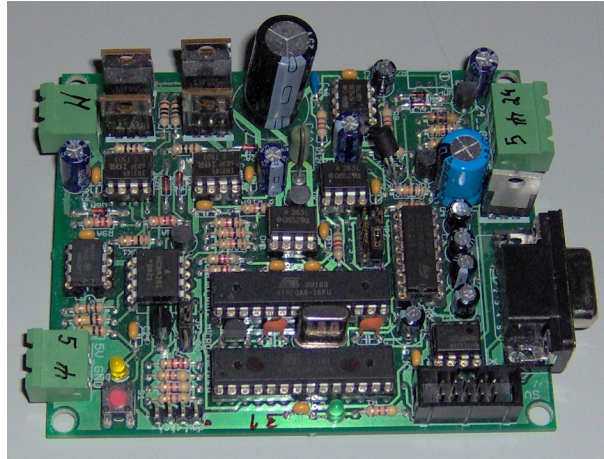


Figura 3.6: Placa controladora dos motores do robô

Estas placas PCB, vulgarmente designadas de *driver*, efectuam o controlo de potência dos motores através de uma ponte em H de transístores. Estes módulos possuem também microcontroladores que permitem gerir as comunicações com o controlador de nível hierárquico superior e executar tarefas de processamento local. Para além da leitura dos dados dos *encoders* e da actuação nos motores, estes dispositivos implementam controladores de velocidade em malha fechada. Desta forma, a interface com o sistema computacional é bastante facilitada. Para que o robô execute uma dada trajectória, apenas é necessário enviar as velocidades de referência desejadas para cada roda. A actuação nos motores é processada pelo controlador correspondente.

### 3.2.3.1 Dimensionamento dos controladores

A arquitectura do sistema controlado é ilustrada na figura 3.7.

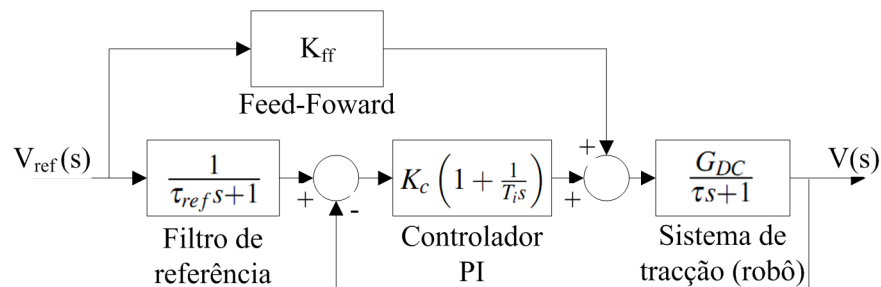


Figura 3.7: Arquitectura do conjunto controladores e sistema de tracção

Para além de um controlador do tipo PI, é ainda implementado um filtro do sinal de referência e um controlo *Feed-Forward*. O objectivo do filtro de referência é anular o zero introduzido pelo controlador, pelo que o valor do seu pólo terá de ser igual ao zero do controlador. O controlador *Feed-Forward* faz com que a resposta seja mais rápida e aperfeiçoada sem influência ao nível da

estabilidade [3]. A função de transferência do sistema, que relaciona a velocidade  $v$  de uma das rodas com a sua referência  $v_{ref}$ , é dada por:

$$\frac{V(s)}{V_{ref}(s)} = \frac{\frac{1}{\tau_{ref}s+1} * K_c \left(1 + \frac{1}{T_i s}\right) * \frac{G_{DC}}{\tau s+1}}{1 + \frac{K_c \left(1 + \frac{1}{T_i s}\right) * G_{DC}}{\tau s+1}} = \frac{\left(\frac{K_c G_{DC}}{T_i \tau}\right)}{s^2 + \left(\frac{1+K_c G_{DC}}{\tau}\right)s + \frac{K_c G_{DC}}{T_i \tau}} \quad (3.1)$$

Antes de se proceder ao dimensionamento dos controladores dos motores de tracção, é necessário obter um modelo da sua dinâmica. Para isso aplicou-se um degrau de tensão a cada um dos motores e registou-se a evolução temporal das velocidades dos mesmos. A resposta do sistema foi aproximada por um modelo de 1ª ordem.

A função de transferência de um sistema de 1ª ordem é dada por:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{G_{DC}}{1 + \tau s} \quad (3.2)$$

Sendo a equação diferencial associada:

$$\tau \dot{Y}(t) + Y(t) = G_{DC} * X(t) \quad (3.3)$$

Onde  $G_{DC}$  representa o ganho DC e  $\tau$  a constante de tempo em regime permanente.

A resposta do sistema a uma entrada em degrau  $u(t)$  é dada por:

$$v(t) = G_{DC} * \left(1 - e^{-t/\tau}\right) * u(t) \quad (3.4)$$

Para estimar os parâmetros  $G_{DC}$  e  $\tau$  que caracterizam o modelo, aplicou-se um método de aproximação de mínimos quadrados aos dados obtidos na resposta ao degrau. Deste processo resultaram os parâmetros dos sistemas que modelizam a dinâmica de cada uma das rodas motrizes. De seguida, efectuou-se a média entre os valores de  $G_{DC}$  e  $\tau$  estimados para a roda direita e esquerda, obtendo-se os valores exibidos na tabela 3.1. Na figura 3.8 é apresentada a evolução temporal das velocidades em cada roda após aplicação de um degrau de tensão, bem como a estimativa dada pelo modelo aproximado de 1ª ordem.

Tabela 3.1: Valores dos parâmetros do modelo dinâmico de 1ª ordem do robô

Ganho DC - $G_{DC}$	4,514
Constante de tempo - $\tau$ (s)	0,288

De seguida, foram calculados os parâmetros dos controladores,  $K_c$  e  $T_i$ , de acordo com o protótipo de *Bessel*. Os pólos da função de transferência, apresentada na equação 3.1, foram calculados a partir dos pólos de *Bessel* para um tempo de estabelecimento de 1 s. Na tabela 3.2 encontram-se os valores calculados dos parâmetros dos controladores.



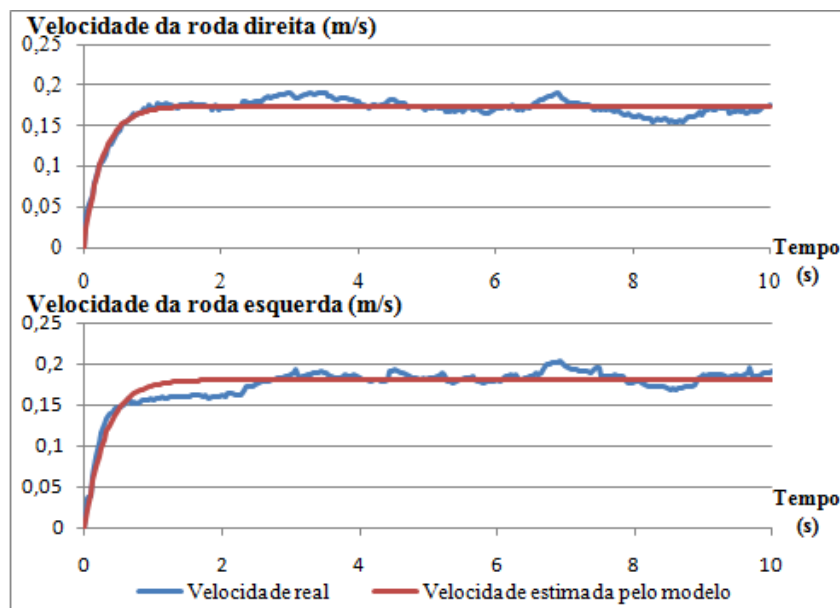


Figura 3.8: Resposta ao degrau e estimativa dada pelo modelo aproximado de 1ª ordem

Tabela 3.2: Valores dos parâmetros dos controladores dos motores de tracção

Ganho do controlador - $K_c$	0,296
Tempo integral - $T_i$ (s)	0,212
Constante de tempo do filtro de referência - $\tau_{ref}$	0,212

### 3.2.4 Sistema de visão artificial

O sistema de visão artificial é composto por uma câmara e respectiva estrutura de suporte e condicionamento da cena. O seu objectivo é, conforme referido anteriormente, adquirir imagens do pavimento, onde estarão colocados marcadores visuais que auxiliam a localização do veículo. O método de localização assente nestes dispositivos será apresentado na secção 4.2, pelo que aqui interessa apenas referir os equipamentos e as escolhas tomadas para aquisição da informação visual.

#### 3.2.4.1 Escolha da câmara e lente

Optou-se por utilizar uma câmara USB monocromática, resolução 1024x768, da marca *Imaging Source*, disponível no laboratório (semelhante à da figura 3.9). Esta escolha levou em conta a existência de *driver's* e de alguns exemplos de aplicações deste tipo de câmaras associados ao grupo de investigação do laboratório onde se realizou este projecto. Além disso, não é necessária alimentação externa, uma vez que o cabo de dados USB já fornece a potência requerida.

A opção por uma câmara a preto e branco levou em consideração o tipo de informação que se pretende retirar. Conforme apresentado em 4.2, a forma e tonalidade do marcador foram escolhidas de modo a este ser facilmente distinguível da cena. O processo de identificação do



Figura 3.9: Câmera USB monocromática da *Imaging Source*

marcador não necessita de informação de cor, baseando-se apenas na detecção do contorno do mesmo. Além disso, é costume as câmaras a cores de baixo custo utilizarem apenas um sensor CCD em cada pixel, em vez dos três sensores CCD correspondentes às cores vermelha, verde e azul, distribuídos segundo o padrão *Bayer* (ver figura 3.10).

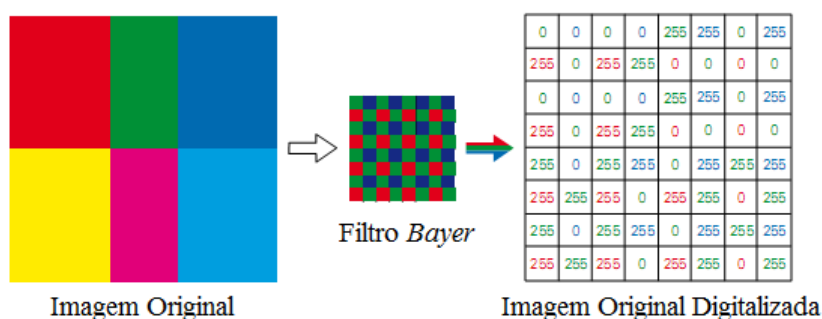


Figura 3.10: Padrão de *Bayer* [8]

Como cada pixel adquire apenas a intensidade de uma cor, a conversão para RGB é calculada com base nos valores dos pixels vizinhos, através de, por exemplo, uma interpolação bilinear (ilustrada na figura 3.11). Esta operação comporta-se como um filtro passa-baixo, suavizando as transições de cor, o que poderia dificultar a detecção da orla do marcador. Além disso, as imagens obtidas por este processo tendem a ser mais ruidosas do que as imagens fornecidas por câmaras a preto e branco [39]. Daí, a opção por uma câmara monocromática apresentar-se mais vantajosa.

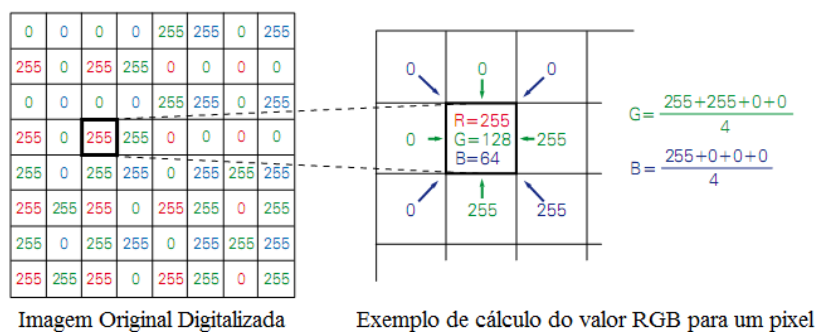


Figura 3.11: Exemplo de conversão do formato *Bayer* para RGB [8]

Após a escolha da câmara torna-se necessário escolher uma lente que permita a aquisição de toda a área de interesse aproveitando ao máximo a resolução da câmara. A zona onde se poderá

detectar os marcadores encontra-se delimitada pela estrutura do veículo, conforme se exemplifica na figura 3.12.

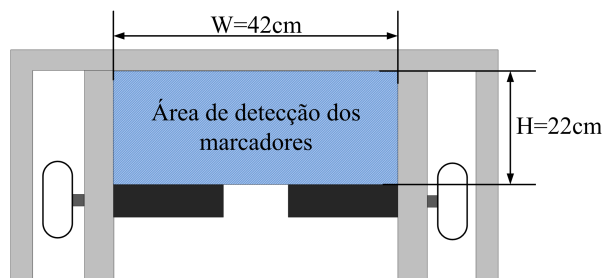


Figura 3.12: Dimensões da cena a captar pela câmara

É necessário portanto assegurar que a área de 42 cm de largura e de 22 cm de altura seja adquirida na totalidade, para que o marcador seja visível ao sistema de detecção durante o maior intervalo de tempo possível. A câmara foi instalada de maneira a que a largura da imagem a captar ficasse associada à zona mais larga do CCD. A altura a que a câmara pode ser colocada do solo é igualmente imposta pelas características estruturais do veículo. Para que a linha de visão da câmara seja perpendicular ao plano da imagem a captar, a máxima distância a que esta pode ser instalada do solo é de 39 cm. Este valor constitui a distância de trabalho,  $W_d$ , utilizada neste projecto.

A partir dos dados fornecidos pelo fabricante fica-se a conhecer a largura e altura do CCD, respectivamente,  $CCD_W=4.8\text{mm}$  e  $CCD_H=3.6\text{mm}$ . Com estes dados é possível calcular o valor máximo da distância focal,  $f$ , tanto para a largura,  $W$ , como para a altura,  $H$ , da imagem que se pretende captar, através das seguintes equações:

$$f_W = \frac{W_d \times CCD_W}{W + CCD_W} \quad (3.5)$$

$$f_H = \frac{W_d \times CCD_H}{H + CCD_H} \quad (3.6)$$

Para os valores em questão, os resultados obtidos foram:

- $f_W = 4.37 \text{ mm}$
- $f_H = 6.23 \text{ mm}$

A máxima distância focal da lente é imposta pelo valor mais baixo entre  $f_W$  e  $f_H$ , neste caso  $f_W$ . Assim, ao se adquirir uma lente deve-se escolher, de entre os valores padrão disponíveis, uma com distância focal o mais próximo desse valor mínimo, mas sempre inferior a este.

Durante a realização deste projecto foi utilizada uma lente com zoom disponível no laboratório, e ajustada para um valor próximo de  $f_W$ . No entanto, caso fosse necessário adquirir uma lente nova seria aconselhável optar por uma sem zoom, pois são mais leves, de menores dimensões e, principalmente, mais baratas. O valor padrão admissível mais próximo da distância focal pretendida é de 4 mm. Esta distância focal é das mais baixas disponíveis comercialmente,

pelo que seria necessário optar por uma lente do tipo *wide-angle*, isto é, uma lente que permite captar imagens de grandes dimensões a curtas distâncias. A diferença de valor entre a lente escolhida e  $f_w$  conduziria a um desperdício da resolução da imagem, que poderia ser compensado pela diminuição da distância de trabalho (instalando a câmara mais próxima do solo).

### 3.2.4.2 Condicionamento da cena

Outro factor de extrema importância é o condicionamento da iluminação da cena. A imagem capturada pela câmara USB monocromática mapeia em cada pixel a informação referente à luminosidade captada pelo sensor correspondente. A alteração das condições de iluminação pode fazer com que, para a mesma imagem, os pixels da *frame* adquirida apresentem valores de luminosidade díspares, dificultando por isso a funcionalidade dos algoritmos de processamento e recolha da informação pretendida. No caso de um robô móvel, a navegação pode efectuar-se em ambientes com características de iluminação muito diversas, pelo que o desenvolvimento de sistemas robustos baseados em visão artificial deve levar em conta este aspecto.

Para condicionamento da iluminação optou-se por conceber e instalar uma caixa acrílica opaca, na zona dianteira do veículo onde serão detectados os marcadores visuais. Após instalação verificou-se que, apesar do limite inferior desta estrutura estar ao mesmo nível da parte inferior do veículo, isto é, a apenas 2 centímetros do solo, a presença de luz exterior ainda interferia numa porção da região a capturar pela câmara. Assim, optou-se por colocar um material maneável, neste caso cartolina, em toda a periferia da base da caixa acrílica de modo a prolongar o efeito da mesma. Esta solução eliminou por completo a presença de luz exterior na cena a captar.

De seguida, surge a necessidade de iluminação artificial, que, neste projecto, foi suprida por duas placas PCB com duas séries de 19 LED's cada. Após a realização de testes concluiu-se que a melhor opção para distribuir a luz de modo uniforme por todo o plano do pavimento era recorrer a iluminação indirecta, instalando as placas no interior da caixa acrílica de cada um dos lados da câmara, conforme se mostra na figura 3.13.

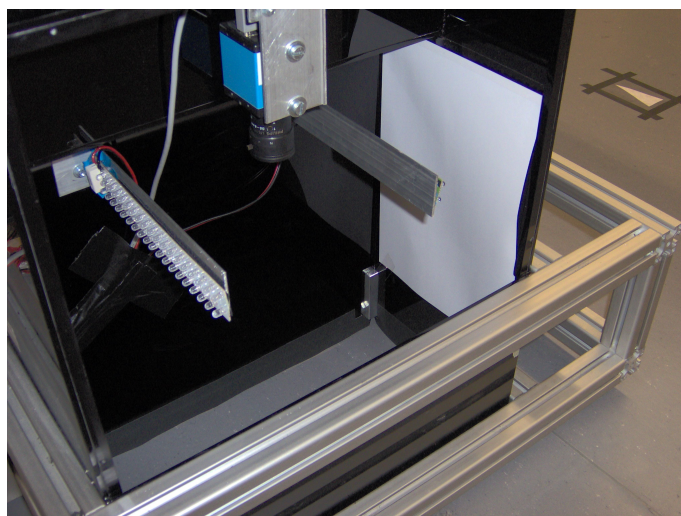


Figura 3.13: Posicionamento da iluminação artificial e da câmara

### 3.2.5 Laser Range Finder

Neste projecto, o *Laser Range Finder* será instalado na zona frontal do veículo e utilizado para detecção de obstáculos, bem como para reconhecimento de marcadores com características de relevo, usados para localização absoluta do robô.

De modo geral, um *Laser Range Finder* permite, através de um varrimento, determinar a distância ao objecto mais próximo segundo um grande número de sentidos direccionais, fornecendo assim um mapa bidimensional do ambiente que o rodeia com elevada precisão. Após cada medida de distância, o equipamento roda o sensor segundo uma fracção da sua resolução angular e volta a efectuar uma leitura, repetindo este ciclo até percorrer toda a gama de medição, completando assim um varrimento. Porém, existe uma zona na qual, devido a limitações físicas e mecânicas, o sensor não é capaz de tirar dados, designada de *dead zone*. Os dados fornecidos pelo *Laser Range Finder* mapeiam num gráfico polar cada uma das medidas tiradas, ao longo de uma circunferência de  $360^\circ$  cuja origem é a posição onde se encontra o dispositivo, conforme ilustra a figura 3.14. Este varrimento é realizado constantemente, permitindo a obtenção periódica de informação actualizada.

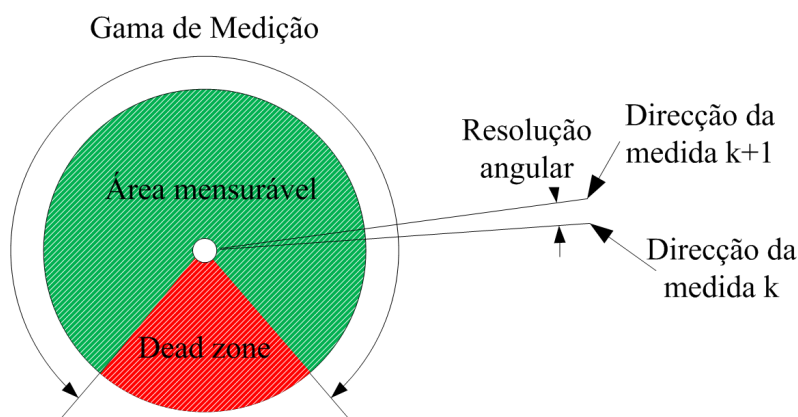


Figura 3.14: Varrimento efectuado por um *Laser Range Finder*

O dispositivo *Laser Range Finder* utilizado é o URG-04LX-UG01 da marca *Hokuyo*, representado na figura 3.15, e foi desenvolvido segundo as normas aplicáveis em ambientes industriais. Este equipamento usa um *laser* de luz infravermelha com comprimento de onda de 785nm de classe de segurança 1. A sua área de varrimento é um semicírculo de  $240^\circ$  e com alcance compreendido entre 6 cm e 4 m. A resolução angular do varrimento é de aproximadamente  $0.36^\circ$ , o que significa que, ao longo da gama de detecção do sensor, são tiradas 683 medidas de distância diferentes.

A medição de distância é devolvida com uma resolução de 1 mm, ao passo que a precisão é de  $\pm 3$  cm, para medidas até 1 metro, e de  $\pm 3$  por cento da medida para distâncias superiores. O diâmetro do raio *laser* emitido é inferior a 20mm até a uma distância de 2m e tem uma divergência máxima de 40mm a uma distância de 4m.



Figura 3.15: Hokuyo URG-04LX-UG01 - *Laser Range Finder* utilizado

O princípio de medida da distância baseia-se no cálculo da diferença de fase entre pulsos enviados, o que permite obter medidas estáveis com influência mínima da cor e da opacidade do objecto [40].

A troca de dados com o *Laser Range Finder* efectua-se através de um protocolo de comunicação baseado em caracteres ASCII. Assim, para que a aplicação de controlo receba os dados do *laser*, é necessária uma implementação em *software* capaz de comunicar com este e de fornecer a devida interface.

### 3.2.6 Sistema computacional

O sistema computacional é responsável por todo o processamento de alto nível necessário ao controlo, localização e navegação do robô. De modo que, para além de suprir as interfaces necessárias aos vários sistemas que constituem o robô, deve-se garantir que este dispositivo apresenta poder computacional suficiente para executar as aplicações no tempo de ciclo desejado.

Durante a realização deste projecto, as aplicações de *software* desenvolvidas foram executadas num computador portátil, com as seguintes características: Intel Core2 Duo CPU T7500 @ 2,20 GHz, 2 GB RAM. Existe ainda a possibilidade de se utilizar um PC industrial para o efeito.

### 3.2.7 Sistema de alimentação

A energia requerida pela maioria dos sistemas do robô é provida por um par de baterias de ácidos, cíclicas, de tensão nominal de 12 Volt ligadas em série, fornecendo uma tensão de 24 Volt. As baterias adquiridas permitem ciclos contínuos de carga e descarga mostrando-se adequadas para aplicações industriais. A tensão de alimentação fornecida é ainda adaptada através de conversores DC-DC para outros níveis de tensão, de modo a suprir as diferentes necessidades conforme segue:

- 24 V  $\Rightarrow$  Circuito de potência dos drivers dos motores;
- 24 V - 12 V  $\Rightarrow$  Circuito de iluminação do sistema de visão artificial;
- 24 V - 6 V  $\Rightarrow$  Concentrador (*Hub*) USB;
- 24 V - 5 V  $\Rightarrow$  Circuitos de comando dos drivers;

### 3.3 Plataforma de Software e Comunicações

O desenvolvimento de todo o software, incluindo a implementação dos algoritmos de reconhecimento de marcadores, localização e detecção de obstáculos, foi realizado sobre a plataforma *Lazarus*. Trata-se de um ambiente de desenvolvimento integrado *opensource* baseado em *FreePascal* e largamente utilizado nos trabalhos a decorrer no laboratório onde foi realizado o projecto. Existe ainda um componente para esta plataforma que permite a utilização de câmaras *Firewire* e USB, desenvolvido pela equipa de futebol robótico 5DPO. Além disso, a aplicação de controlo do robô teve por base o material herdado de outro projecto realizado sobre a mesma plataforma.

Durante a realização deste projecto foram utilizadas três aplicações de software, que serão brevemente apresentadas de seguida. O envio de informação para a aplicação central de controlo é feito através do protocolo UDP, um dos mais utilizados em aplicações de tempo real. A sua principal vantagem reside na velocidade de envio dos pacotes de rede. Este protocolo não é orientado à conexão e não garante a entrega dos pacotes. No entanto, deve-se notar que, por exemplo, a câmara capta imagens a 15 FPS, pelo que serão enviados 15 pacotes por segundo para a rede sempre que o robô se encontrar sobre um marcador. Como a actualização e envio dos dados é cíclica, a eventual perda de um pacote não é crítica, pois um novo pacote será entregue num intervalo de tempo reduzido. Aliás, durante testes experimentais verificou-se que quando o robô passa por um marcador à velocidade de, por exemplo, 30 cm/s, o sistema detecta e envia 9 medidas consecutivas (ver secção 6.1.3).

A aplicação de interface com o *Laser Range Finder* (ver figura 3.16) foi desenvolvida pelo Prof. Dr. Paulo Costa. Esta aplicação corre a cada 100 ms, isto é, o intervalo de tempo a que ciclicamente o *laser* efectua um varrimento. Este programa implementa o protocolo de comunicação com o *laser* e sempre que recebe uma leitura reencaminha os dados para a aplicação de controlo através de um pacote UDP.

Para além disso, a aplicação de interface com o *laser* incorpora algoritmos de processamento capazes de identificar a presença de marcadores em relevo do género de códigos de barras. Sempre que um conjunto de segmentos válidos seja detectado numa leitura, um pacote UDP adicional é enviado para a aplicação de controlo. Este pacote contém a informação da largura e do centro de cada um dos segmentos identificados em relação ao referencial do *laser*. Na secção 4.3 serão dadas mais informações sobre o formato do marcador utilizado e a metodologia adoptada para localização absoluta do robô.

A interface gráfica da aplicação desenvolvida para detecção dos marcadores visuais encontra-se na figura 3.17. Esta aplicação implementa os algoritmos apresentados no capítulo 5. A execução das tarefas de processamento é desencadeada após a recepção de uma nova *frame* da câmara, o que ocorre ciclicamente a cada 66 ms. No final de um ciclo, caso seja detectado um marcador, é enviado um pacote UDP com as coordenadas do centro e orientação do marcador em relação ao referencial do robô, bem como a percentagem de incerteza da detecção e o código identificador do marcador (ver tabela 3.3).



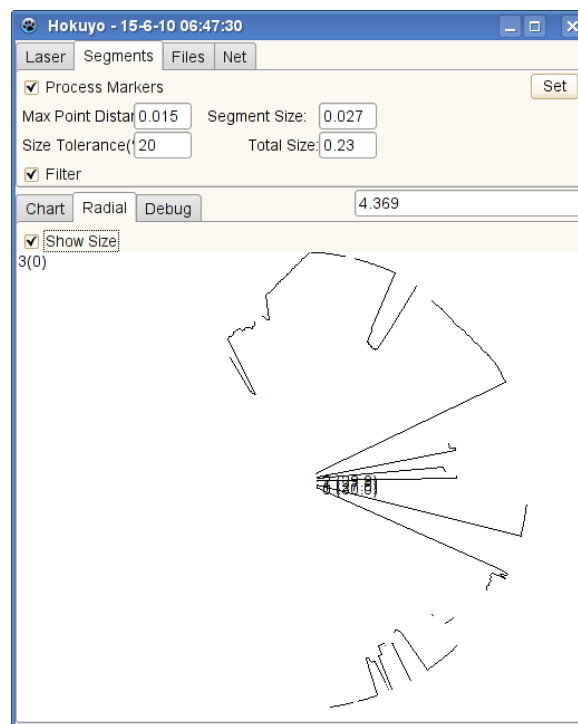
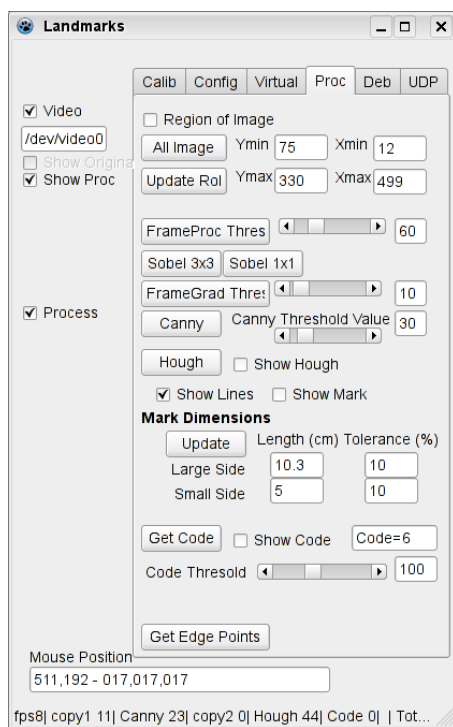
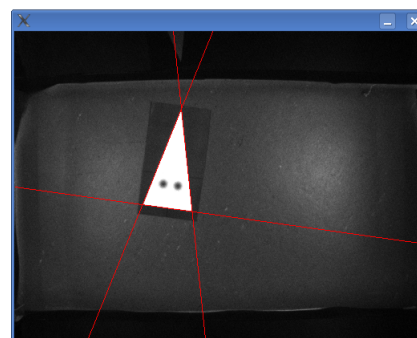


Figura 3.16: Layout da aplicação de interface com o *Laser Range Finder*



(a) Interface principal



(b) Visualização em tempo real da imagem da câmara

Figura 3.17: Interface gráfica da aplicação de detecção dos marcadores visuais



Tabela 3.3: Formato da mensagem enviada por UDP pela aplicação de detecção de marcadores visuais

#	uncertainty	*	code	*	x	*	y	*	orientation	*
---	-------------	---	------	---	---	---	---	---	-------------	---

Finalmente, a outra aplicação desenvolvida é responsável pelo controlo total do veículo. A sua interface gráfica, ilustrada na figura 3.18, permite definir os vários parâmetros de controlo e visualizar a movimentação do robô. Esta aplicação é executada periodicamente a cada 40 ms. A recepção de mensagens das restantes aplicações é vista como um evento externo. Os novos dados recebidos são processados, guardados em memória e utilizados no ciclo de processamento seguinte.

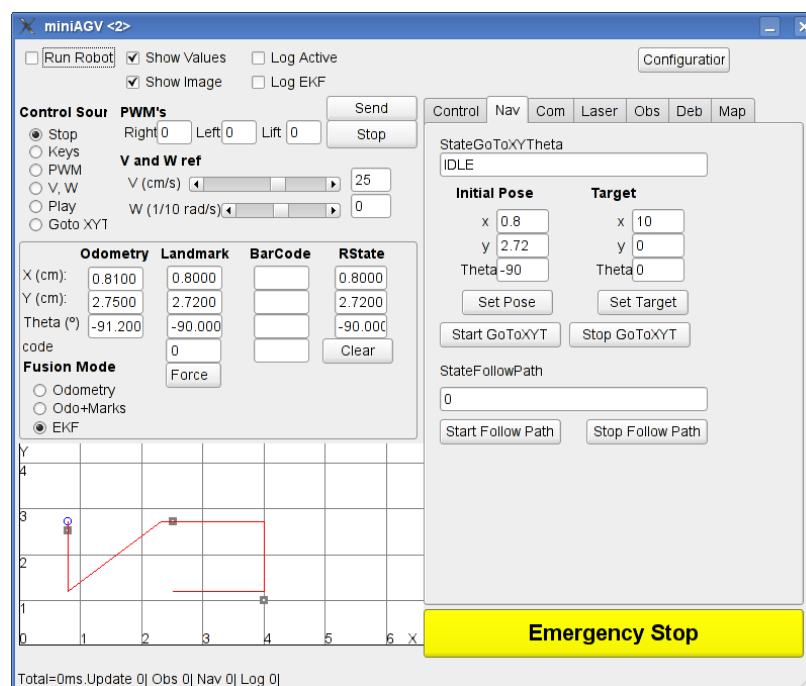


Figura 3.18: Interface gráfica da aplicação de controlo do robô

### 3.4 Detecção de Obstáculos

Nesta secção serão apresentadas as técnicas implementadas para detecção de obstáculos.

A principal abordagem faz uso das medidas de distância fornecidas pelo *Laser Range Finder*. Instalado na zona frontal do veículo, este dispositivo permite detectar a presença de obstáculos no interior da sua área de medida. Como são conhecidos os ângulos segundo os quais se realiza cada uma das medidas e o valor de distância das mesmas, as coordenadas  $(x, y)$  dos pontos detectados ficam totalmente definidas.

Tendo em conta o posicionamento dos obstáculos em relação ao robô foram definidas duas zonas distintas de actuação, conforme se ilustra na figura 3.19.

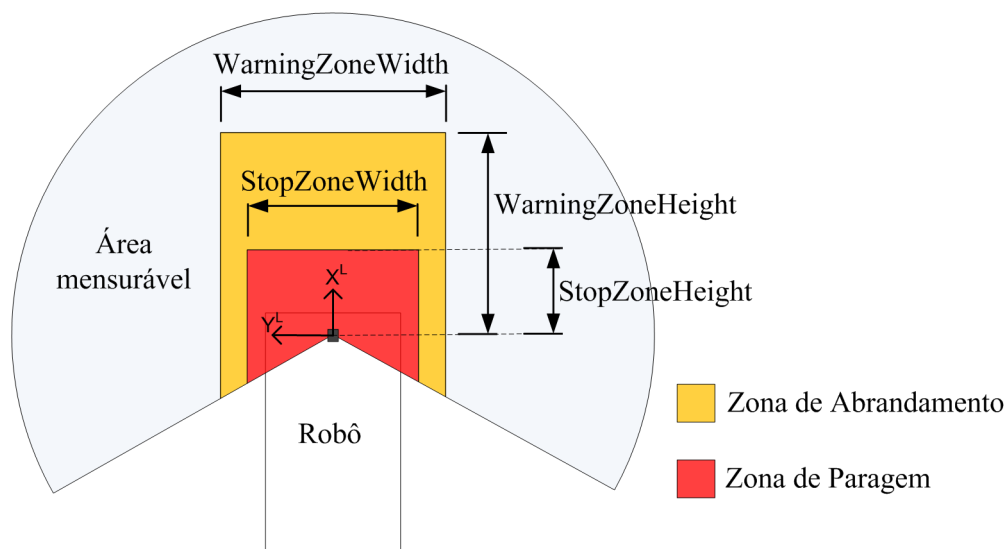


Figura 3.19: Zonas de segurança para detecção de obstáculos

Sempre que um determinado número de pontos medidos pelo *laser* pertence à zona de paragem, é imediatamente enviado um comando para que o robô pare. O robô permanece imóvel até à desobstrução da zona definida como de paragem. No caso de se detectar um impedimento na zona de abrandamento e não na de paragem, é permitido ao veículo continuar a navegação. No entanto, neste caso, a velocidade do robô é limitada a um valor inferior ao nominal, que o permita parar em segurança se necessário.

O robô retorna ao seu estado de operação normal sempre que as zonas anteriormente definidas estejam desimpedidas. Seria igualmente possível fazer o robô aguardar por uma confirmação para reiniciar o movimento após uma paragem.

As zonas de segurança apresentadas são flexíveis, podendo ser alteradas durante o funcionamento do robô.

### 3.4.1 Técnicas Complementares

Infelizmente, a abordagem anteriormente apresentada não evita todo o tipo de colisões. Sendo assim, foram implementadas técnicas complementares de segurança.

#### 3.4.1.1 Limitação da corrente máxima nos motores

Em caso de colisão ou bloqueio do robô durante o seu movimento, a corrente exigida pelos motores tenderá a aumentar e exceder o seu valor nominal. As placas controladores dos motores possuem já um mecanismo que permite estimar o valor da corrente fornecida,  $I_a$ . Consiste numa resistência,  $R$ , em série com a ponte em H de transístores. Durante a condução, a corrente  $I_a$  provoca uma queda de tensão em  $R$ . O valor médio desta tensão,  $V_{adc}$ , é obtido após a passagem por um filtro passa-baixo e adquirido por um conversor analógico-digital do micro-controlador da

placa, após isolamento da tensão. O valor estimado de  $I_a$  pode ser calculado através da seguinte equação:

$$I_a = \frac{V_{adc}}{R} * \frac{255}{PWM} \quad (3.7)$$

$PWM$  é o valor de *duty – cycle* numa escala entre 0 e 255. A resistência  $R$  tem um valor baixo, 0,22  $\Omega$ , para minimizar a dissipação de energia.

Sempre que, para valores não nulos de  $PWM$ , a corrente medida  $I_a$  ultrapasse um dado valor máximo, o robô é imobilizado e permanece neste estado até desbloqueio na aplicação de controlo.

### 3.4.1.2 Modelização da dinâmica do conjunto controlador e sistema de tracção

Outra técnica consiste em comparar a resposta do robô a um comando de velocidade com a que seria de esperar por um modelo. Para isso foi determinada uma aproximação de 1ª ordem à função de transferência apresentada em 3.1, isto é:

$$G^{apr}(s) = \frac{G_{DC}^{apr}}{1 + \tau^{apr}s} \quad (3.8)$$

Os parâmetros  $G_{DC}^{apr}$  e  $\tau^{apr}$ , o ganho DC e a constante de tempo em regime permanente, foram determinados através de um método de aproximação de mínimos quadrados aplicado à resposta do sistema a um degrau de velocidade.

A versão discreta desta função de transferência pode ser representada por:

$$G^{apr}(z) = \frac{V_{est}(z)}{v_{ref}(z)} = \frac{K_2^{disc}}{z - K_1^{disc}} \quad (3.9)$$

Com:

$$K_1^{disc} = e^{-T/\tau^{apr}} \quad (3.10)$$

$$K_2^{disc} = G_{DC}^{apr} * \left(1 - e^{-T/\tau^{apr}}\right) \quad (3.11)$$

$T$  é o período de amostragem usado.

A estimativa da velocidade instantânea de cada uma das rodas motrizes do robô,  $v_{est}(k)$ , pode ser calculada, em cada ciclo de controlo, pela seguinte equação às diferenças obtida a partir da equação 3.9:

$$v_{est}(k) = K_1^{disc} * v_{est}(k-1) + K_1^{disc} * v_{ref}(k-1) \quad (3.12)$$

Assim, temos um modelo de primeira ordem que relaciona a referência de velocidade enviada para o controlador do motor de cada roda motriz com a respectiva velocidade instantânea esperada. Este modelo é de fácil implementação, pois apenas requer que seja guardado em memória o valor de velocidade estimado no instante anterior. As velocidades instantâneas medidas e as

calculadas pelo modelo, para cada roda motriz, são ciclicamente comparadas. Caso se verifique uma discrepância entre estes valores superior a um dado limite, pressupõe-se que o robô colidiu, pelo que é dada ordem de paragem.

Os valores obtidos dos parâmetros deste modelo aproximado de 1<sup>a</sup> ordem encontram-se na tabela 3.4.

Tabela 3.4: Valores dos parâmetros do modelo discreto de 1<sup>a</sup> ordem do conjunto sistema de tracção e controlador

$K_1^{disc}$	0,726
$K_2^{disc}$	0,274

## Capítulo 4

# Localização

A capacidade de se auto-localizar é um dos requisitos fundamentais para a autonomia de um robô móvel. O objectivo da localização é determinar a posição  $x$ ,  $y$  e a orientação  $\theta$  do veículo em relação a um referencial externo.

Os métodos de localização de robôs móveis habitualmente utilizados podem ser resumidos a essencialmente dois tipos de abordagens: medidas relativas e medidas absolutas do ambiente. Neste projecto, a pose<sup>1</sup> do robô é ciclicamente estimada a partir da informação fornecida pela odometria (ver secção 4.1) e corrigida sempre que um método de localização baseado em medidas absolutas esteja disponível. Os dois métodos de localização absoluta desenvolvidos fazem uso de marcadores passivos e são apresentados nas secções 4.2 e 4.3. A fusão das estimativas oriundas dos diferentes métodos de localização é tratada na secção 4.4.

### 4.1 Localização relativa - Odometria

A odometria é um dos métodos mais simples e mais amplamente utilizados para estimação momentânea da pose de um robô. Sendo um método de localização relativa, a sua utilização em aplicações práticas tem, quase sempre, por objectivo o fornecimento de informação actualizada em tempo real da posição e orientação do veículo durante os intervalos de tempo em que outro método de localização absoluta é periodicamente executado. O deslocamento do robô a partir de uma posição inicial conhecida é obtida pela integração dos movimentos lineares de cada uma das rodas.

As principais vantagens da odometria estão associadas ao baixo custo de implementação, à elevada taxa de amostragem permitida e à boa precisão alcançada a curto prazo. A principal restrição prende-se com a acumulação ilimitada de erros ao longo da navegação, fruto da integração dos erros com que as medidas do movimento das rodas são afectadas. Este facto faz com que a estimativa da pose do robô fornecida por este método se afaste dos seus valores reais,

---

<sup>1</sup>O termo pose refere-se à posição cartesiana e orientação do robô em relação a um referencial externo.

impossibilitando que a sua utilização seja bem sucedida durante um período alargado de tempo, sem recorrer a um método de localização absoluta.

Na secção seguinte, 4.1.1, é apresentado o modelo da odometria. Os seus erros sistemáticos e calibração são abordados na secção 4.1.2. Na secção 4.1.3 são mencionados os erros não sistemáticos e apresenta-se um método de quantificar a incerteza da estimativa da pose do robô, através de um modelo de erro da odometria.

### 4.1.1 Modelo da Odometria

Na figura 4.1 encontra-se representado o modelo do robô diferencial no sistema coordenado global  $0X^0Y^0Z^0$  (a omissão do eixo  $Z^0$  deve-se à consideração de que o robô se desloca no plano definido por  $X^0Y^0$ ). A pose do robô é definida pela sua posição  $(x_r^0, y_r^0)$  e orientação,  $\theta_r^0$ , em relação ao sistema coordenado global.

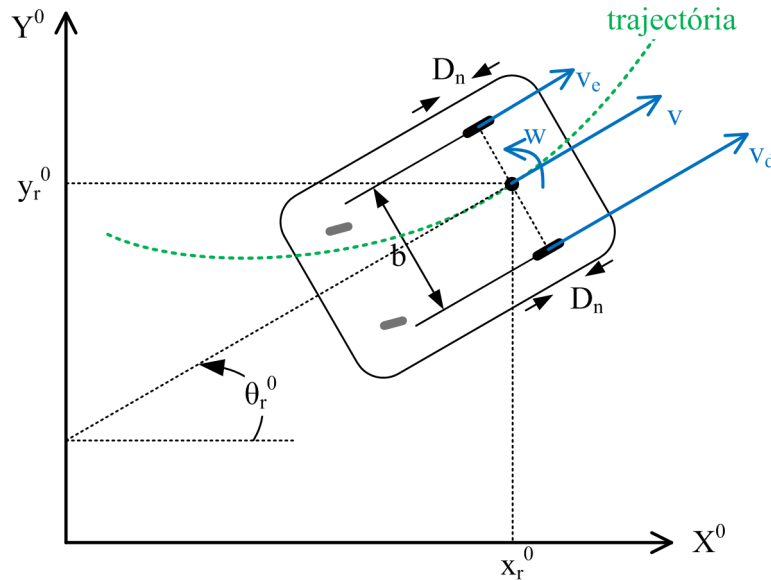


Figura 4.1: Robô diferencial no sistema coordenado global

O modelo cinemático do robô diferencial [41] pode ser representado pelo seguinte sistema de equações:

$$\frac{d}{dt} \begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} v_r(t) * \cos(\theta_r(t)) \\ v_r(t) * \sin(\theta_r(t)) \\ \omega_r(t) \end{bmatrix} \quad (4.1)$$

As variáveis de entrada deste sistema são as velocidades linear,  $v_r(t)$ , e angular,  $\omega_r(t)$ , do veículo que podem ser obtidas a partir das velocidades lineares de cada roda motriz:

$$v_r(t) = \frac{v_d(t) + v_e(t)}{2} \quad (4.2)$$

$$\omega_r(t) = \frac{v_d(t) - v_e(t)}{b} \quad (4.3)$$

$b$  representa a distância entre as rodas dianteiras do robô no ponto de contacto com o solo.

A discretização por diferenças centradas do modelo cinemático, permite encontrar a expressão de actualização da pose do robô a cada instante de amostragem:

$$\begin{bmatrix} x_r(i+1) \\ y_r(i+1) \\ \theta_r(i+1) \end{bmatrix} = \begin{bmatrix} x_r(i) + \Delta d(i) * \cos(\theta_r(i) + \frac{\Delta\theta(i)}{2}) \\ y_r(i) + \Delta d(i) * \sin(\theta_r(i) + \frac{\Delta\theta(i)}{2}) \\ \theta_r(i) + \Delta\theta(i) \end{bmatrix} \quad (4.4)$$

A aproximação da orientação do veículo durante o período de amostragem pelo ângulo intermédio,  $\theta_r(i) + \frac{\Delta\theta(i)}{2}$ , procura minimizar o erro introduzido pela discretização. Os deslocamentos linear e angular do robô,  $\Delta d(i)$  e  $\Delta\theta(i)$ , são calculados, de modo análogo às equações 4.2 e 4.3, através de:

$$\Delta d(i) = \frac{\Delta d_d(i) + \Delta d_e(i)}{2} \quad (4.5)$$

$$\Delta\theta(i) = \frac{\Delta d_d(i) - \Delta d_e(i)}{b} \quad (4.6)$$

O deslocamento linear de cada roda durante um período de amostragem,  $\Delta d_d(i)$  e  $\Delta d_e(i)$ , é estimado a partir do número de impulsos,  $N_{d/e}(i)$ , gerados pelo *encoder* correspondente, isto é:

$$\Delta d_{d/e}(i) = c_m * N_{d/e}(i) \quad (4.7)$$

com,

$$c_m = \frac{\pi D_n}{n R_e} \quad (4.8)$$

O termo  $c_m$  representa o factor de conversão que relaciona o número de impulsos do *encoder* com o deslocamento horizontal;  $D_n$  corresponde ao diâmetro nominal das rodas;  $R_e$  é a resolução dos *encoder's* e  $n$  a relação da caixa redutora.

#### 4.1.2 Erros sistemáticos e calibração da odometria

Para que o modelo da odometria fique completamente definido, é necessário determinar os parâmetros que lhe estão associados. A resolução dos *encoder's* e a relação da caixa redutora são conhecidos, pelo que resta apenas estimar  $D_n$  e  $b$ .

Começou-se por estimar o diâmetro nominal das rodas, que se assume igual em ambas. Para isso, fez-se o robô percorrer em linha recta uma distância de 6 metros e registou-se o número total de impulsos gerados pelos *encoder's*. Com base nestes dados e nas equações 4.5, 4.7 e 4.8,  $D_n$  pode ser facilmente obtido.

Finalmente, para estimar a distância entre rodas no ponto de contacto com o solo,  $b$ , aplicou-se o mesmo módulo de velocidade a ambas as rodas mas de sentidos opostos, ficando o robô

a rodar sobre si próprio. Ao fim de 10 voltas completas, registou-se novamente o número total de impulsos gerados pelos *encoder's* durante este procedimento. A informação disponível até o momento permite, através da equação 4.6, obter a estimativa da distância entre rodas,  $b$ .

Os valores dos parâmetros  $D_n$  e  $b$  obtidos nestas duas experiências encontram-se na tabela 4.1.

Tabela 4.1: Valores dos parâmetros do modelo de odometria do robô

Diâmetro das rodas ( $D_n$ )	15,06 cm
Distância entre rodas ( $b$ )	61,43 cm

Este método de calibração, apesar da sua simplicidade, permite obter uma estimativa do diâmetro médio das rodas e da distância entre elas mais precisa do que a que seria obtida por medição. Para além destes dois factores, existem outras fontes de erros sistemáticos, como, por exemplo, o desalinhamento das rodas e a diferença de diâmetro das mesmas, cujo efeito no cálculo da odometria foi considerado suficientemente baixo para requerer uma calibração mais complexa. De salientar ainda que a rigidez das rodas do robô em estudo é elevada e que não proporciona, por isso, a sua deformação com o aumento da carga, diminuindo assim a influência negativa que este efeito poderia ter na estimativa da pose do veículo dada pela odometria.

A compensação mais precisa dos erros sistemáticos obrigaria à utilização de métodos de calibração mais completos. Um deles, conhecido como *UMBMark*, é apresentado por *Borenstein et. al.* [42] e permite estimar e compensar separadamente os erros devidos à diferença no diâmetro das rodas e à incerteza da distância entre os pontos de contacto destas com o solo. Informações mais detalhadas sobre este método de calibração podem ser consultadas na referência [42].

### 4.1.3 Erros não sistemáticos e modelo de erro da odometria

Diferentemente dos erros sistemáticos, existem outros erros imprevisíveis que resultam principalmente da interacção do robô com o ambiente à sua volta. Estes erros, habitualmente designados de aleatórios ou não sistemáticos, são causados, por exemplo, pelo resvalamento das rodas (associado a pisos escorregadios ou aceleração excessiva), irregularidades no pavimento ou presença de objectos inesperados no percurso.

A própria resolução finita dos *encoder's*, bem como a discretização e aproximações introduzidas na formulação do modelo da odometria, podem ser consideradas fontes de erros não sistemáticos [41]. No entanto, os seus efeitos podem ser bastante atenuados ao se utilizar uma frequência de leitura dos dados dos *encoder's* suficientemente elevada para a velocidade a que o robô se desloca. No caso particular do robô em estudo, o comportamento dinâmico das rodas livres, do tipo “castor”, gera forças internas difíceis de prever, condicionando a correcta interpretação do real deslocamento do robô.

Apesar de não se poder compensar previamente este tipo de erros, é possível estimar a incerteza associada à pose do robô calculada pela odometria. Com esse objectivo foi implementado um modelo estocástico do erro da odometria, conforme apresentado por *Crowley* [43], que se passa a descrever de seguida.



Ao sistema de equações 4.4 é adicionado, a cada instante de amostragem, uma parcela relativa aos erros da odometria, obtendo-se:

$$\begin{bmatrix} x_r(i+1) \\ y_r(i+1) \\ \theta_r(i+1) \end{bmatrix} = \begin{bmatrix} x_r(i) + \Delta d(i) * \cos\left(\theta_r(i) + \frac{\Delta\theta(i)}{2}\right) \\ y_r(i) + \Delta d(i) * \sin\left(\theta_r(i) + \frac{\Delta\theta(i)}{2}\right) \\ \theta_r(i) + \Delta\theta(i) \end{bmatrix} + W(i) \quad (4.9)$$

Considera-se que o ruído  $W(i)$  possui distribuição gaussiana de média nula e matriz de covariância  $Q(i)$ :

$$E[W(i)] = 0 \quad (4.10)$$

$$E[W(i) * W(i)^T] = Q(i) \quad (4.11)$$

$E[.]$  representa o valor esperado.

Pressupondo que os ruídos que afectam as estimativas de  $x_r$ ,  $y_r$  e  $\theta_r$  são independentes entre si, a matriz  $Q(i)$  é uma matriz diagonal:

$$Q(i) = \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix} \quad (4.12)$$

Os elementos da diagonal principal são definidos da seguinte forma:

$$Q_{11}(i) = K_{uu} |\Delta d(i) * \cos(\Delta\theta(i))| \quad (4.13)$$

$$Q_{22}(i) = K_{uu} |\Delta d(i) * \sin(\Delta\theta(i))| \quad (4.14)$$

$$Q_{33}(i) = K_{u\theta} |\Delta d(i)| + K_{\theta\theta} |\Delta\theta(i)| \quad (4.15)$$

$K_{uu}$  é o coeficiente que relaciona o erro da odometria segundo  $d(i)$  (translação do robô) em relação a  $\Delta d(i)$ ,  $K_{u\theta}$  é o coeficiente que relaciona o erro da odometria segundo  $\theta(i)$  (rotação do robô) em relação a  $\Delta d(i)$  e  $K_{\theta\theta}$  o coeficiente que relaciona o erro da odometria segundo  $\theta(i)$  em relação a  $\Delta\theta(i)$ .

A matriz de covariância do erro da estimativa da pose do robô,  $C(i)$ , é definida por:

$$C(i) = \begin{bmatrix} Var(x_r(i)) & Covar(x_r(i), y_r(i)) & Covar(x_r(i), \theta_r(i)) \\ Covar(x_r(i), y_r(i)) & Var(y_r(i)) & Covar(y_r(i), \theta_r(i)) \\ Covar(x_r(i), \theta_r(i)) & Covar(y_r(i), \theta_r(i)) & Var(\theta_r(i)) \end{bmatrix} \quad (4.16)$$

$Var(.)$  e  $Covar(.)$  representam, respectivamente, a variância e covariância.

A matriz de covariância da estimativa da pose fornecida pela odometria é actualizada através de:

$$C(i+1) = \Phi^*(i) * C(i) * \Phi^{*T}(i) + Q(i) \quad (4.17)$$

$\Phi^*(i)$  corresponde à matriz de transição de estado e é dada, em cada instante de amostragem, por:

$$\Phi^*(i) = \begin{bmatrix} 1 & 0 & -\Delta d(i) * \sin\left(\theta(i) + \frac{\Delta\theta(i)}{2}\right) \\ 0 & 1 & \Delta d(i) * \cos\left(\theta(i) + \frac{\Delta\theta(i)}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

De modo a estimar os parâmetros  $K_{uu}$ ,  $K_{u\theta}$  e  $K_{\theta\theta}$  do modelo de erro apresentado, foram realizados dois ensaios experimentais [44].

Em primeiro lugar fez-se o robô percorrer, em linha recta, uma distância de 4 metros segundo o eixo coordenado  $x^0$  e, no final do percurso, calculou-se o erro da odometria através do sistema de localização baseado em marcadores visuais. A partir dos dados de 20 ensaios, calculou-se  $K_{uu}$  e  $K_{u\theta}$ , definidos como [44]:

$$K_{uu} = \frac{Var(x)}{med(x_{total})} \quad (4.19)$$

$$K_{u\theta} = \frac{Var(\theta)}{med(x_{total})} \quad (4.20)$$

$Var(x)$  e  $Var(\theta)$  representam, respectivamente, a variância do erro da medida de  $x$  e de  $\theta$ , ao passo que  $med(x_{total})$  corresponde à média das distâncias totais percorridas.

Na segunda experiência registou-se o erro do sistema de odometria após uma rotação do robô sobre si próprio. Foram realizados novamente 20 ensaios, permitindo estimar  $K_{\theta\theta}$  [44]:

$$K_{\theta\theta} = \frac{Var(\theta)}{med(\theta_{total})} \quad (4.21)$$

O termo  $med(\theta_{total})$  representa a média do deslocamento angular realizado.

Os valores obtidos para os parâmetros do modelo de erro da odometria encontram-se na tabela 4.2.

Tabela 4.2: Valores dos parâmetros do modelo de erro da odometria do robô

$K_{uu}$	1,417E-4 m <sup>2</sup> /m
$K_{u\theta}$	3,215E-3 rad <sup>2</sup> /m
$K_{\theta\theta}$	2,782E-4 rad <sup>2</sup> /rad

De salientar ainda que os valores destes parâmetros depende não só do robô, mas também do ambiente em que este se desloca. Assim, caso as experiências realizadas fossem efectuadas em pisos mais irregulares ou rugosos, seria de esperar estimativas de  $K_{uu}$ ,  $K_{u\theta}$  e  $K_{\theta\theta}$  de valor superior.

Estes parâmetros definem o modelo de erro apresentado e representam a confiança que temos no sistema da odometria.

## 4.2 Localização absoluta - Marcadores visuais

Este método de localização baseia-se num sistema que detecta figuras geométricas previamente dispostas sobre o pavimento com o objectivo de fornecer uma estimativa precisa da pose do robô.

Na verdade, existem já vários trabalhos de investigação que propõem sistemas de localização de robôs móveis baseados em visão e marcadores artificiais passivos. Por exemplo, *Taeyeon Kim* [45] utiliza marcadores pretos com forma rectangular fixados no tecto em intervalos regulares. *Sooyong Lee* [46] também utiliza marcadores artificiais no tecto, mas com uma forma geométrica mais elaborada. O marcador é um triângulo equilátero com nove sectores, três dos quais servem para identificar o marcador na imagem e obter a orientação do mesmo em relação ao robô. Os restantes seis sectores permitem a identificação distinta de até 64 marcadores. Além disso, a câmara utilizada é equipada com iluminação infravermelha e os marcadores são revestidos com uma película reflectora própria de modo a minimizar a perturbação da iluminação ambiente. Em [47] é apresentada uma técnica para localização global utilizando apenas um marcador visual colocado numa parede.

Porém, em ambientes industriais, não é fácil garantir um ambiente limpo, pelo que a utilização de uma câmara a apontar para o tecto poderia obrigar a constante manutenção. Além disso, quanto mais complexa for a forma geométrica do marcador, mais rapidamente uma degradação impossibilita a sua identificação. Assim, optou-se por escolher um marcador em forma de triângulo isósceles (ver a figura 4.2) colocado no pavimento, o que permite definir a posição cartesiana  $x$ ,  $y$  do robô, bem como a sua orientação, após a detecção de um único elemento.

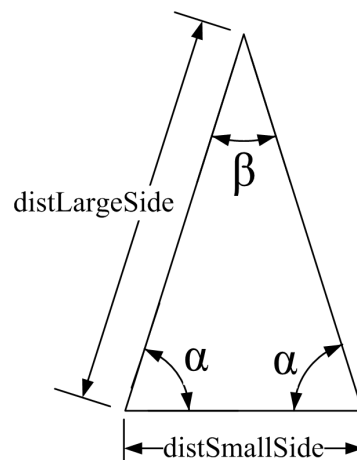


Figura 4.2: Forma geométrica do marcador visual escolhido

A detecção do marcador (cuja dimensão utilizada se encontra na tabela 4.3) faz uso do sistema de visão artificial, referido na secção 3.2.4. Os algoritmos de processamento e obtenção da pose do marcador em relação ao referencial do robô encontram-se no capítulo 5. De salientar que este sistema pode ser facilmente reajustado para diferentes dimensões do triângulo.

Tabela 4.3: Dimensões do marcador visual utilizado

<i>distLargeSide</i>	10,3 cm
<i>distSmallSide</i>	5 cm
$\beta$	28,07°
$\alpha$	75,97°

Conforme mencionado na secção 5.6, é possível ainda associar ao marcador um código identificador compreendido entre 0 e 63. No entanto, caso se usasse apenas marcadores com códigos, o número máximo de elementos utilizáveis ficaria limitado. Por isso, foram desenvolvidas duas abordagens à correcção da pose pelos marcadores visuais.

A primeira abordagem faz uso de marcadores não codificados, isto é, sem qualquer tipo de mancha. Estes elementos podem ser utilizados repetidamente e são todos codificados com o número 0, reservado para o efeito. Assim, sempre que um marcador com código nulo é detectado, consulta-se um mapa de modo a identificar o marcador mais próximo da posição actual estimada do robô. Esta abordagem pode ser aplicada em zonas onde a estimativa da pose do robô permanece suficientemente precisa durante o trajecto entre marcadores.

A segunda abordagem pretende que a pose do robô seja determinada sem qualquer conhecimento prévio da localização do veículo. Neste caso, os códigos compreendidos entre 1 e 63 identificam posições únicas no mapa. Quando um marcador com código não nulo é detectado, a pesquisa no mapa de marcadores devolve, caso exista, a sua posição (x, y) e orientação em relação ao referencial global, que é única. Esta abordagem pode ser aplicada, por exemplo, em zonas onde o robô passa periodicamente. O objectivo é evitar que o robô se “perca” por falhar a visualização dos marcadores sem código. Outra aplicação interessante desta abordagem é nos locais de partida, pois permite ao robô iniciar a navegação sem qualquer estimativa prévia da sua localização.

A posição, orientação e codificação de todos os marcadores, face ao referencial global, é guardada num ficheiro *XML*, como exemplificado no seguinte excerto:

```
...
<cuboid title='... '>
  <imovable/>
  <pos x='...' y='...' theta='...'/>
</cuboid>
...
```

Esta metodologia é bastante flexível, pois permite que um utilizador altere a localização dos marcadores sem proceder à modificação do código fonte da aplicação. Sempre que for iniciada a aplicação, a base de dados dos marcadores é carregada com as informações contidas no ficheiro.

A partir da pose do marcador em relação ao referencial do robô,  $(x_m^1, y_m^1, \theta_m^1)$ , e da sua localização no referencial mundo,  $(x_m^0, y_m^0, \theta_m^0)$ , é calculada a pose do robô,  $(x_r^0, y_r^0, \theta_r^0)$ , face ao referencial global (ver figura 4.3):

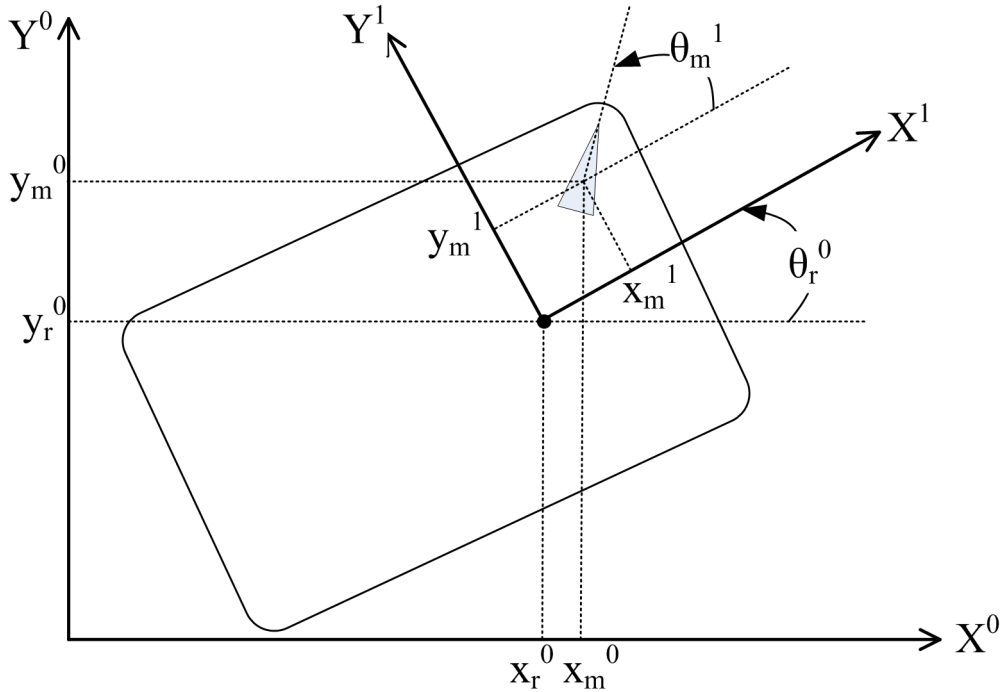


Figura 4.3: Correção da pose do robô a partir de um marcador visual

$$\begin{cases} \theta_r^0 = \theta_m^0 - \theta_m^1 \\ x_r^0 = x_m^0 - (x_m^1 \cos \theta_r^0 - y_m^1 \sin \theta_r^0) \\ y_r^0 = y_m^0 - (y_m^1 \cos \theta_r^0 - x_m^1 \sin \theta_r^0) \end{cases} \quad (4.22)$$

### 4.3 Localização absoluta - Marcadores em relevo

Este método de localização absoluta utiliza marcadores em relevo semelhantes a um código de barras. Estes elementos são detectados através do varrimento executado pelo *Laser Range Finder*. Cada marcador possui 10 posições cujo valor binário é definido como 1 na presença de uma barra e 0 em caso contrário (ver figura 4.4).

As posições 0, 1 e 9 são reservadas para identificação dos extremos do marcador e são sempre atribuídas com valor 1. As restantes posições servem para definir o código do marcador. Genericamente, um código binário com 7 elementos permite gerar até 128 combinações distintas. No entanto, na versão actual da aplicação de interface com o *Laser Range Finder* optou-se por criar as seguintes restrições à criação dos códigos:



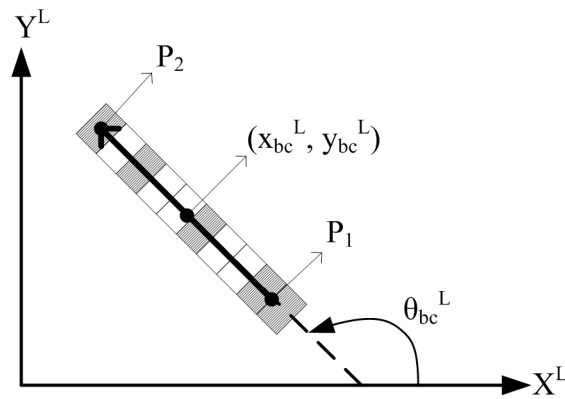


Figura 4.6: Cálculo do centro e da orientação do marcador em relevo no referencial do *laser*

De seguida, calcula-se a pose do marcador no referencial do robô. Como os eixos  $x$  e  $y$  dos referenciais do robô e do *laser* são paralelos, este cálculo equivale, neste caso, a apenas uma translação segundo o eixo  $x$ .

O código do marcador é obtido a partir dos segmentos detectados para além do iniciador e do terminador. Para cada segmento, calcula-se a distância do seu centro ao centro do segmento iniciador, *distToRight*. Para os segmentos de largura unitária, o bit do código a que correspondem é dado por:

$$codeBit = round\left(\frac{distToRight - 1/2 * segmentSize}{segmentSize}\right) \quad (4.25)$$

Onde  $round(.)$  representa o arredondamento para inteiro mais próximo.

No caso de segmentos de largura dupla são incrementados dois bit's do código:

$$codeBit1 = round\left(\frac{distToRight - segmentSize}{segmentSize}\right) \quad (4.26)$$

$$codeBit2 = round\left(\frac{distToRight}{segmentSize}\right) \quad (4.27)$$

Nesta fase, o código final do marcador encontra-se totalmente definido:

$$DetectedBarcode.code = bit1 + 2 * bit2 + 4 * bit3 + 8 * bit4 + 16 * bit5 + 32 * bit6 + 64 * bit7 \quad (4.28)$$

A correcção da pose do robô a partir da pose do marcador processa-se de modo análogo ao apresentado no caso dos marcadores visuais, na secção 4.2.

Na figura 4.7 encontra-se um dos protótipos construídos para uso em testes. Neste caso, a largura de cada barra, *segmentSize*, é de 2,7 cm.

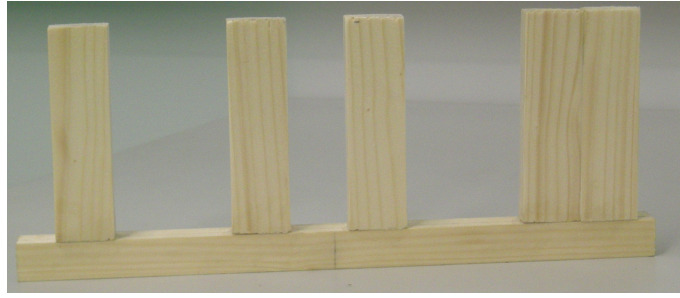


Figura 4.7: Exemplo de um protótipo do marcador de código de barras (código 20)

## 4.4 Fusão sensorial

A fusão da informação proveniente da odometria e dos métodos de localização absoluta foi realizada com recurso a um Filtro de *Kalman* Estendido.

O Filtro de *Kalman* é uma das técnicas mais utilizadas para fusão sensorial. Trata-se de um método matemático que garante uma estimação óptima no sentido de que, caso todos os processos sejam lineares e os respectivos ruídos gaussianos, o filtro minimiza o quadrado dos erros dos parâmetros estimados [9]. O mesmo conceito pode ser aplicado a sistemas não lineares usando a variante do Filtro de *Kalman* anteriormente referida.

O Filtro de *Kalman* Estendido, em cada instante, lineariza a função de transferência em torno do valor do estado estimado e actualiza as estimativas baseado num Filtro de *Kalman* aplicado a esse sistema linear. Apesar de não apresentar garantias teóricas de convergência e de optimalidade, na maioria dos casos, o seu desempenho é muito satisfatório e evita a complexidade do projecto de um filtro óptimo para o sistema não linear em causa.

Uma descrição mais detalhada do Filtro de *Kalman* e do Filtro de *Kalman* Estendido pode ser encontrada, por exemplo, nas referências [48], [49] e [9]. Na secção seguinte é sumariada a implementação do Filtro de *Kalman* Estendido neste projecto.

### 4.4.1 Implementação do Filtro de *Kalman* Estendido

Dado que se pretende uma aproximação ao modelo cinemático do robô móvel apresentado na equação 4.1, parte-se da seguinte descrição dinâmica do sistema em tempo contínuo:

$$\frac{dX(t)}{dt} = f(X(t), u(t_k), t), \quad t \in [t_{k-1}, t_k] \quad (4.29)$$

O estado do sistema,  $X(t)$ , corresponde à pose estimada do robô:

$$X(t) = \begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} \quad (4.30)$$



A entrada do sistema,  $u(t)$ , é a informação de velocidade linear,  $v_r(t)$ , e angular,  $w_r(t)$ , fornecida pela odometria:

$$u(t) = \begin{bmatrix} v_r(t) \\ w_r(t) \end{bmatrix} \quad (4.31)$$

A linearização da dinâmica do sistema em torno de  $t = t_k$ ,  $X(t) = X(t_k)$  e  $u(t) = u(t_k)$  resulta em:

$$A^*(k) = \left. \frac{\partial f}{\partial x} \right|_{\substack{X(t) = X(t_k) \\ u(t) = u(t_k) \\ t = t_k}} = \begin{bmatrix} 0 & 0 & -v_r(t_k) * \sin(\theta(t_k)) \\ 0 & 0 & v_r(t_k) * \cos(\theta(t_k)) \\ 0 & 0 & 0 \end{bmatrix} \quad (4.32)$$

Considerando que os valores permanecem constantes durante cada intervalo de amostragem,  $\Delta T = t_k - t_{k-1}$ , obtemos a matriz de transição de estado:

$$\phi^*(k) = \exp(A^*(k) * \Delta T) = \begin{bmatrix} 1 & 0 & -v_r(t_k) * \sin(\theta(t_k)) * \Delta T \\ 0 & 1 & v_r(t_k) * \cos(\theta(t_k)) * \Delta T \\ 0 & 0 & 1 \end{bmatrix} \quad (4.33)$$

A implementação em tempo discreto do Filtro de *Kalman* Extendido é composta por duas fases distintas, designadas de previsão e actualização. Durante a operação de previsão, é usado o estado do robô no instante anterior para estimar o estado do sistema no instante seguinte. Esta previsão da evolução do sistema utiliza o modelo de odometria do robô, pelo que durante esta fase a covariância da estimativa do estado tenderá a aumentar, reflectindo a diminuição da confiança neste cálculo. Durante a operação de actualização, as medidas efectuadas, isto é, as estimativas do estado do robô obtidas pelos métodos de localização absoluta, são usadas para corrigir a estimativa do estado resultante da fase de previsão. Esta informação extra permite melhorar a precisão da estimativa do estado do sistema e aumentar a confiança nessa informação, pelo que durante esta fase a covariância da estimativa do estado tende a baixar.

Assim, para um dado instante  $k$ , existem dois momentos distintos que serão designados, respectivamente, por  $k^-$  e  $k$ : o primeiro ocorre após a fase de previsão e imediatamente antes da actualização, o segundo corresponde ao momento no qual se dispõe da estimativa final do estado,  $x(k)$ , após a fase de actualização. Deste modo, a estimativa do estado obtida na fase de previsão é designada por  $x(k^-)$ , ao passo que após a fase de actualização passará a ser representada por  $x(k)$ .

- **Condições iniciais:**

Uma vez que a execução do algoritmo do Filtro é recursiva, é necessário definir uma condição inicial para o estado do sistema e para a matriz de covariância associada:

$$E(X(0)) = X_0 \quad (4.34)$$

$$\text{covar}(X(0)) = P_0 \quad (4.35)$$

• **Fase de Previsão:**

1. Estimativa do estado no instante  $t = t_k$ ,  $X(k^-)$ , a partir do estado no instante  $t = t_{k-1}$ ,  $X(k-1)$ , e da entrada  $u(t_k)$ . Este cálculo é efectuado de acordo com o sistema de equações 4.4.
2. Propagação da covariância da estimativa do estado:

$$P(k^-) = \Phi^*(k) * P(k-1) * \Phi^*(k)^T + Q(k) \quad (4.36)$$

$Q(k)$  é a matriz de covariância do ruído do processo e reflecte o rigor do modelo. Esta matriz encontra-se definida pelas equações 4.12 a 4.15 e pelos parâmetros apresentados na tabela 4.2.

• **Fase de Actualização:**

Esta fase é executada apenas quando existem medidas, isto é, dados provenientes dos sistemas de localização absoluta baseados em marcadores.

1. Cálculo do ganho do filtro de *Kalman*:

$$K(k) = P(k^-) * H(k)^T * (H(k) * P(k^-) * H(k)^T + R(k))^{-1} \quad (4.37)$$

A matriz  $H(k)$  relaciona o estado do sistema com as medidas. Como neste projecto os sistemas de localização absoluta fornecem directamente uma estimativa do estado,  $H(k)$  é uma matriz identidade  $3 \times 3$ .

$R(k)$  é a matriz de covariância do ruído das medidas.

A matriz  $R(k)$  associada à medida fornecida pelo sistema de detecção dos marcadores visuais foi obtida experimentalmente. Assumindo que os ruídos que afectam as medidas de cada componente do estado são independentes entre si e de que possuem distribuição gaussiana de média nula, a matriz  $R(k)$  é diagonal:

$$R(k) = \begin{bmatrix} R_{11} & 0 & 0 \\ 0 & R_{22} & 0 \\ 0 & 0 & R_{33} \end{bmatrix} \quad (4.38)$$

Para determinar os parâmetros desta matriz realizaram-se 6 experiências com um marcador em diferentes posições na área de detecção. Para cada uma delas executou-se o algoritmo de localização 128 vezes e calcularam-se os erros das estimativas de  $x$ ,  $y$  e  $\theta$ , bem como as respectivas variâncias. Os parâmetros  $R_{11}$ ,  $R_{22}$  e  $R_{33}$  foram definidos como as variâncias do erro em  $x$ ,  $y$  e  $\theta$  para o pior caso, isto é, o caso para o

qual os valores são mais elevados (ver tabela 4.4). Este ocorreu quando se posicionou o marcador num dos cantos da área de detecção.

Tabela 4.4: Valores dos parâmetros da matriz de covariância do ruído da medida fornecida pelos marcadores visuais

$R_{11}$	1,65E-7 m <sup>2</sup>
$R_{22}$	1,02E-6 m <sup>2</sup>
$R_{33}$	1,50E-4 rad <sup>2</sup>

De forma análoga, a matriz de covariância do ruído das medidas obtidas pelos marcadores em códigos de barras foi definida como diagonal:

$$R^{bc}(k) = \begin{bmatrix} R_{11}^{bc} & 0 & 0 \\ 0 & R_{22}^{bc} & 0 \\ 0 & 0 & R_{33}^{bc} \end{bmatrix} \quad (4.39)$$

Os parâmetros desta matriz foram obtidos da experiência apresentada em 6.3, novamente utilizando os valores associados ao pior caso. Nesta situação, porém, optou-se por duplicar esses valores, de modo a reflectir melhor a incerteza da medida. Em particular, o valor definido para  $R_{22}^{bc}$  foi 10 vezes superior ao máximo obtido das experiências, uma vez que estas foram realizadas com coordenada  $y$  constante.

Tabela 4.5: Valores dos parâmetros da matriz de covariância do ruído da medida fornecida pelos marcadores em código de barras

$R_{11}^{bc}$	1,2E-5 m <sup>2</sup>
$R_{22}^{bc}$	2,2E-5 m <sup>2</sup>
$R_{33}^{bc}$	5,7E-4 rad <sup>2</sup>

## 2. Actualização da covariância do estado:

$$P(k) = (I - K(k) * H(k)) * P(k^-) \quad (4.40)$$

## 3. Actualização do estado:

$$X(k) = X(k^-) + K(k) * (z(k) - h(X(k^-), 0)) \quad (4.41)$$

$z(k)$  é o vector de medidas, obtidas pela detecção dos marcadores, e  $h(X(k^-), 0)$  equivale a  $X(k^-)$ .

Na figura 4.8 resume-se a operação do Filtro de *Kalman* Extendido.

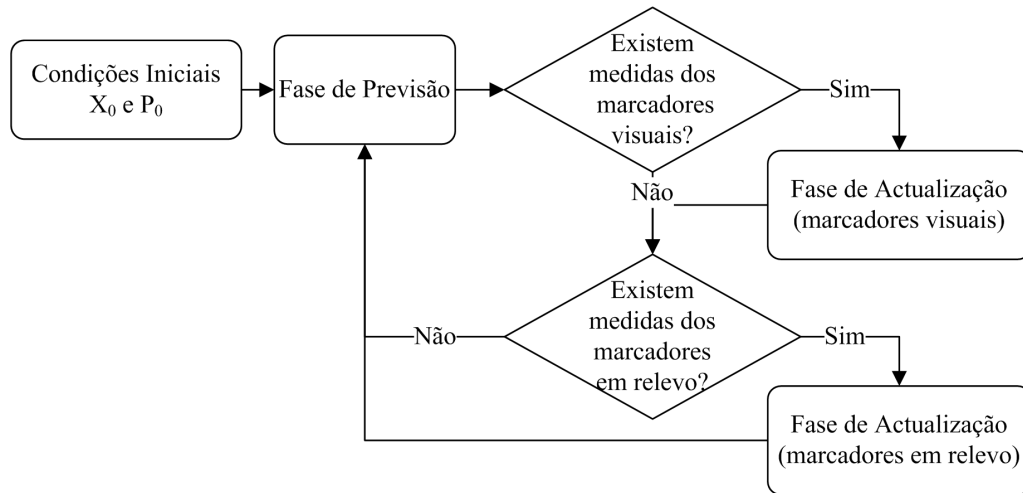


Figura 4.8: Resumo da operação do Filtro de *Kalman* Extendido

## Capítulo 5

# Sistema de detecção de marcadores visuais

Neste capítulo serão apresentados os algoritmos desenvolvidos com vista à identificação dos marcadores visuais e cálculo da sua posição e orientação em relação ao referencial do robô. Esta informação permitirá determinar a pose do veículo no ambiente, conforme apresentado na secção 4.2. Apesar de se aplicar a um caso de estudo particular, os algoritmos desenvolvidos permitem a adaptação ao tamanho do marcador, o ajuste a diferentes condições de luminosidade e degradação do mesmo, bem como a fácil adaptação a diferentes tonalidades do pavimento e do próprio marcador (desde que devidamente contrastantes).

Como referido anteriormente, o sistema de visão artificial é composto por uma câmara USB monocromática. A *frame*  $F$  fornecida é uma matriz de pixeis, em que cada elemento  $F[u,v]$  (ver figura 5.1) mapeia a intensidade de luz captada numa escala de cinzentos, compreendida entre 0 (preto) e 255 (branco).

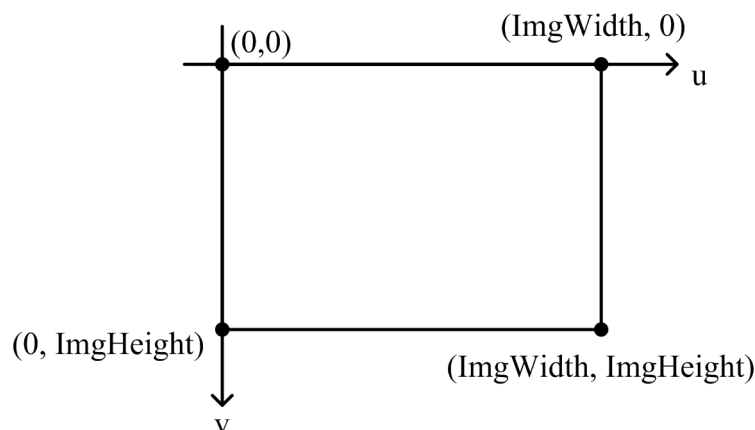


Figura 5.1: Coordenadas e dimensões de uma *frame*

Na figura 5.2 encontra-se um exemplo de uma *frame* captada. É possível constatar que a cena se encontra iluminada de modo bastante uniforme e que a luz exterior apenas incide sobre uma pequena faixa na periferia limítrofe da caixa. Mesmo quando a intensidade da iluminação exterior é elevada, como acontece na zona representada na parte inferior da imagem, a sua influência não se faz sentir no interior da região de interesse, assinalada na figura com um rectângulo a vermelho. Esta região corresponde à zona de detecção dos marcadores que, conforme esperado, não preenche toda a imagem, uma vez que o quociente entre a sua largura e a sua altura é superior à mesma relação do CCD da câmara. Sendo assim, os algoritmos de processamento de imagem são aplicados apenas a esta porção da *frame*, o que resulta em ganhos no tempo de processamento.

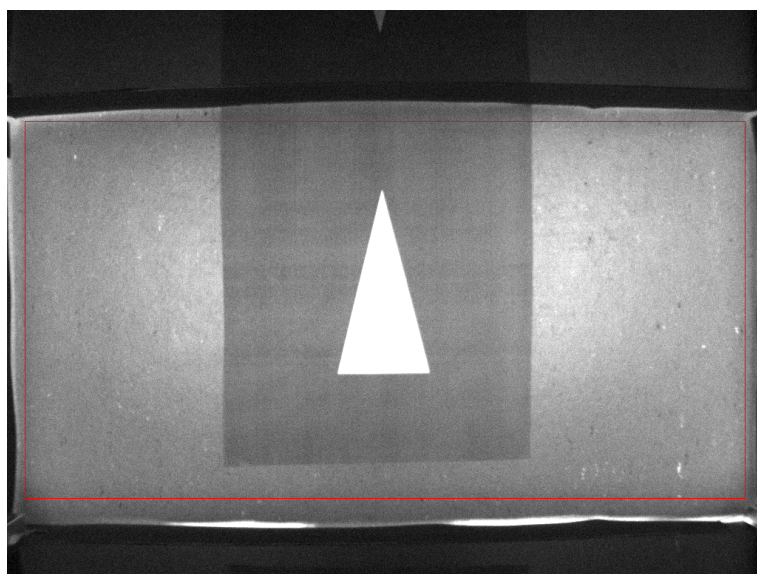


Figura 5.2: Exemplo de uma *frame* captada com indicação da região de interesse

Um factor cujos efeitos se deve ter em atenção é a distorção em barril. A figura 5.3 mostra a imagem captada de uma grelha semelhante a um tabuleiro de xadrez. É possível constatar algum encurvamento das linhas do padrão que, na realidade, são rectas. Este efeito faz com que, por exemplo, os quadrados na zona periférica da grelha sejam percebidos na *frame* com dimensões diferentes dos que se encontram na zona central, quando na realidade não o são, sendo que a variação máxima verificada não ultrapassa os 10 por cento. Apesar disso, e tendo em mente os resultados práticos obtidos, optou-se por não realizar a compensação deste efeito, o que simplifica o algoritmo e reduz o processamento a efectuar. Porém, será necessário avaliar esta opção, comparando os resultados obtidos quando o marcador surge em diferentes zonas da imagem.

Como a detecção dos marcadores necessita de ser realizada mesmo que o robô esteja em movimento, é fundamental garantir que todo o processamento se realize durante um tempo inferior ao da visibilidade total do marcador na área reservada para o efeito. A aplicação de software que implementa a detecção de marcadores é coordenada pela recepção de *frames* da câmara, cujo *frame rate* escolhido foi de 15 FPS. Isto significa que o tempo de processamento do algoritmo está restrito a 66 milissegundos.

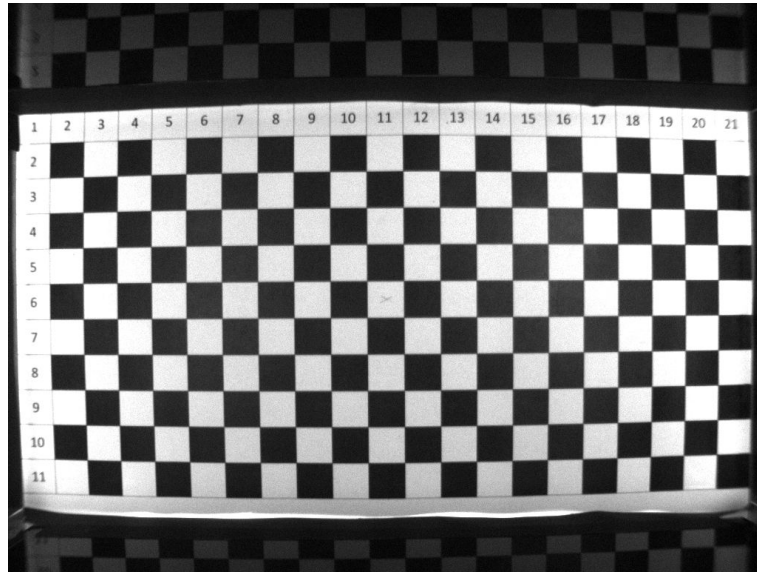


Figura 5.3: Grelha usada para avaliação do efeito da distorção em barril

## 5.1 Detecção da orla do marcador

Após a recepção de uma nova *frame*, o primeiro passo do algoritmo consiste na detecção da orla do marcador. Estas zonas caracterizam-se por uma elevada variação nos níveis de cinzento, próprias do contraste existente entre o marcador de tom claro e o pavimento de cor mais escura, logo com nível de cinzento mais baixo. No entanto, é bom lembrar que o algoritmo não coloca nenhuma restrição quanto à cor do marcador. O único requisito óbvio é que o marcador se destaque do restante do pavimento, pelo que a utilização de, por exemplo, um marcador escuro sobre um fundo de tom claro seria igualmente aceitável.

A detecção de orlas pode ser realizada recorrendo a operadores de gradiente que aproximam a derivada na direcção em que são aplicados, devolvendo um valor elevado quando na presença de um ponto de fronteira e um valor mais baixo em zonas homogéneas. O primeiro operador testado foi o de *Sobel*, que usa duas máscaras 3x3 para calcular o gradiente segundo a direcção horizontal e vertical, cujas matrizes associadas são respectivamente:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (5.2)$$

A magnitude e orientação (no sentido anti-horário em relação ao eixo horizontal  $u$ ) do gradiente são aproximados respectivamente por:

$$|G| \cong \sqrt{G_x^2 + G_y^2} \quad (5.3)$$

$$|\theta_{grad}| \cong \arctan2\left(\frac{G_y}{G_x}\right) \quad (5.4)$$

Após a aplicação deste operador a toda a região de interesse da imagem, obtêm-se duas matrizes: *FrameGrad* e *FrameTheta*, cujo valor em cada elemento  $(u,v)$  corresponde, respectivamente, ao cálculo de  $|G|$  e de  $|\theta_{grad}|$  para o pixel com o mesmo índice,  $F[u,v]$ .

De seguida é necessário seleccionar os pontos de orla que serão usados para identificação dos segmentos do marcador. Para isso procede-se à binarização da matriz *FrameGrad*, ou seja, transforma-se cada elemento de uma escala de cinzentos para um valor binário. A função de binarização pode ser descrita como:

$$FrameGrad_{bin}[u,v] = \begin{cases} 0 & \text{se } FrameGrad[u,v] < threshold \\ 1 & \text{se } FrameGrad[u,v] \geq threshold \end{cases} \quad (5.5)$$

Onde *FrameGrad<sub>bin</sub>* indica a representação binária de *FrameGrad* e *threshold* é o limiar de binarização, abaixo do qual cada elemento passa a ter valor 0 e a partir do qual passa a ter o valor 1. A figura 5.4 apresenta o resultado do processo de binarização após aplicação do operador de *Sobel* sobre uma *frame*.

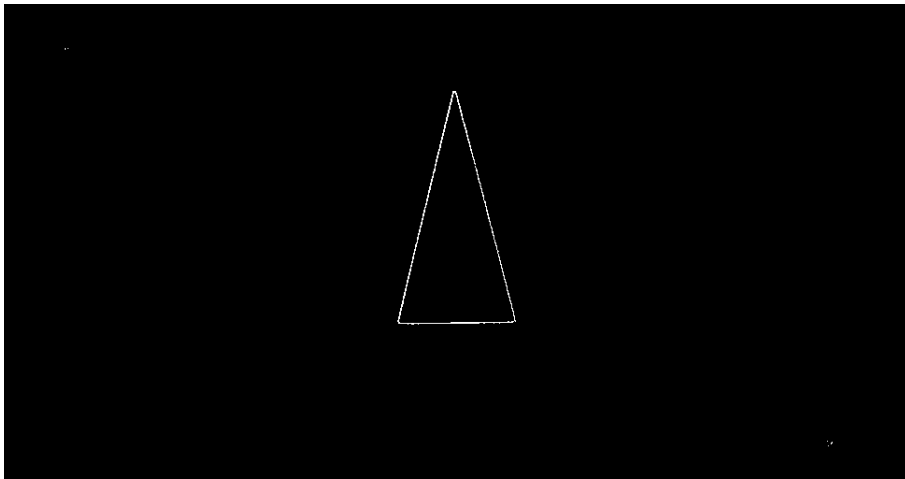


Figura 5.4: Resultado do processo de binarização após aplicação do operador de *Sobel* sobre uma *frame*

O outro operador testado aproxima o gradiente pela derivada de primeira ordem, utilizando apenas os quatro vizinhos mais próximos. O processo de obtenção de uma representação binária



dos pontos de orla é em tudo semelhante ao anteriormente descrito, exceptuando-se o cálculo do gradiente segundo as direcções horizontais e verticais, cujas matrizes associadas passam a ser:

$$G_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.6)$$

$$G_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.7)$$

ou, de um modo simplificado:

$$G_x = F[u+1, v] - F[u-1, v] \quad (5.8)$$

$$G_y = F[u, v+1] - F[u, v-1] \quad (5.9)$$

Na figura 5.5 está representado o resultado binarizado da aplicação deste operador sobre a mesma *frame* usada na figura 5.4.

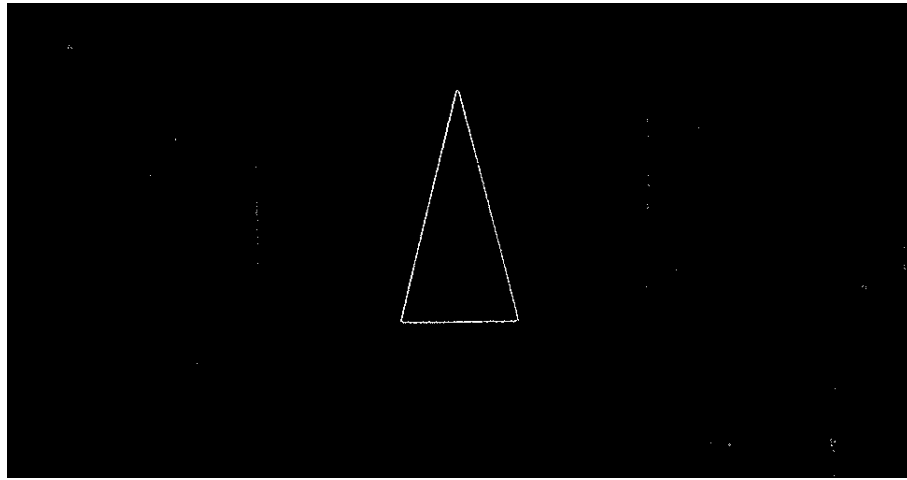


Figura 5.5: Resultado do processo de binarização após aplicação do operador de derivada de primeira ordem sobre uma *frame*

A visualização da matriz  $FrameGrad_{bin}$  permite ajustar o valor de *threshold* a utilizar, de modo a que o contorno do marcador seja totalmente identificado com o mínimo de ruído, isto é, com a detecção do menor número possível de pontos que na realidade não pertencem à orla do marcador. Comparando as figuras 5.4 e 5.5 nota-se que o resultado após aplicação de operador de *Sobel* é menos ruidoso do que o obtido com o operador de primeira ordem. No entanto, como o operador de *Sobel* necessita de avaliar um maior número de pixels vizinhos, os custos computacionais associados aos acessos à memória provocam um forte agravamento do tempo de processamento total. Assim, optou-se por realizar o cálculo do gradiente apenas pela avaliação

dos quatro pixels vizinhos mais próximos, utilizando o operador de primeira ordem descrito. Uma alternativa para diminuir o ruído após a detecção de orlas seria aumentar o valor de *threshold* utilizado na binarização. No entanto, é necessária cautela no ajuste deste valor, pois ao se diminuir o número de pontos assinalados com elevado gradiente, alguns pontos da orla do marcador também podem ser eliminados.

Após este ajuste verifica-se que a orla detectada possui espessura, em termos de pixels, variável e, na maioria dos pontos, não unitária. A aplicação do algoritmo de identificação dos segmentos do contorno sobre esta estrutura de dados obrigaria à avaliação de um número elevado de pontos, com as respectivas implicações no tempo de processamento requerido. Portanto, o próximo passo consiste em descartar os pontos que não correspondem ao máximo local e, por isso, não fornecem informação relevante quanto à localização precisa da orla.

Com esse objectivo foi implementado o processo de supressão de pontos não máximos (*Non-Maximum Supression*), conforme proposto no conhecido algoritmo de detecção de contornos desenvolvido por *John Canny* e que leva o seu nome [50]. O primeiro passo trata de relacionar a direcção do gradiente com uma direcção que possa ser seguida na imagem. Ora, para cada pixel da imagem existem apenas quatro direcções diferentes que descrevem os seus vizinhos: 0 graus (horizontal), 45 graus (diagonal positiva), 90 graus (vertical) e 135 graus (diagonal negativa) [51]. A orientação do gradiente em cada pixel,  $|\theta_{grad}|$ , obtida através da equação 5.4, é convertida para o valor discreto da região à qual pertence, conforme apresentado na figura 5.6.

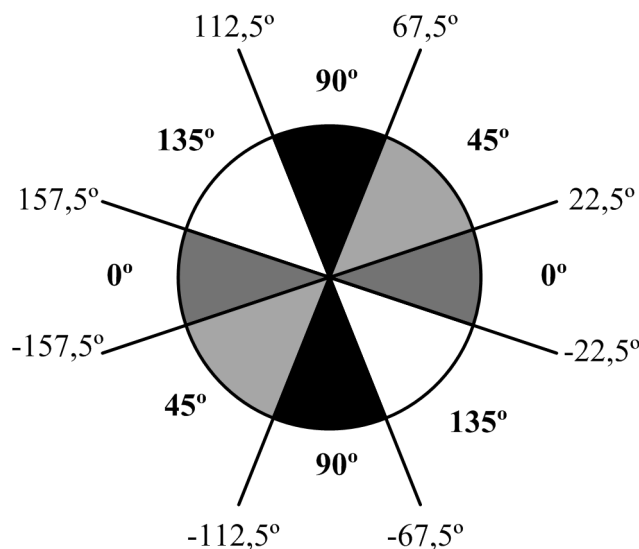


Figura 5.6: Discretização da orientação do gradiente

De seguida, cada um dos pixels assinalados em  $FrameGrad_{bin}$  como pertencente à orla é revisitado e o seu valor de gradiente,  $|G|$ , comparado com os dois pixels adjacentes segundo a direcção do gradiente. Caso o valor de gradiente do pixel em análise não seja superior ao dos seus dois vizinhos, o seu valor na nova representação binária dos pontos de orla,  $FrameCanny_{bin}$ , é colocado a zero, pois não corresponde a um máximo local. Com este processo preserva-se apenas

os pontos correspondentes ao máximo local, ou pico, da orla, reduzindo a espessura de todos os contornos detectados para um pixel (ver figura 5.7).

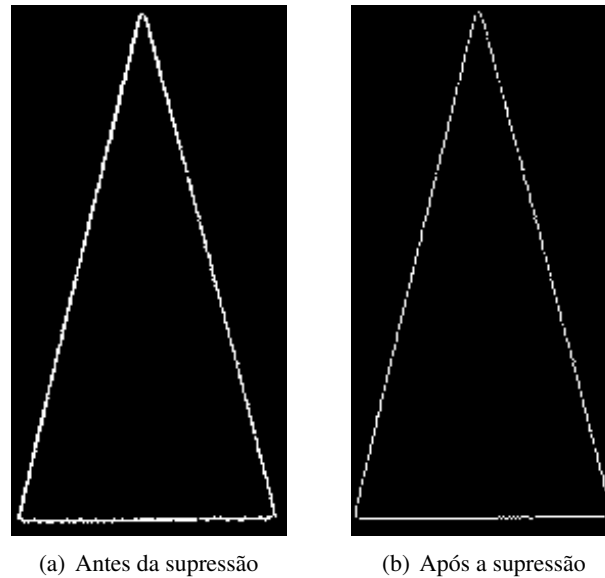


Figura 5.7: Representação binária dos pontos de orla antes e após a supressão dos não máximos locais

## 5.2 Detecção dos segmentos de recta do marcador

Após a obtenção dos pontos de orla detectados na região de interesse da imagem é necessário avaliar a possível presença de um marcador e, em caso afirmativo, caracterizar cada um dos segmentos de recta que compõem o seu contorno. Para isso foi utilizada a transformada de *Hough* [52]. Esta é uma técnica largamente usada no processamento de imagens digitais com o objectivo de detectar formas geométricas como rectas, círculos e elipses.

Em termos simples, a transformada de *Hough* consiste em transformar o espaço cartesiano da imagem num outro espaço de representação. Após a transformação de todos os pontos de orla detectados, a pesquisa de rectas dá-se no espaço transformado e, realizando a transformação inversa, obtêm-se as expressões analíticas dessas rectas na representação original, isto é, no espaço cartesiano da imagem.

Na detecção de linhas, o espaço transformado é uma representação dos parâmetros das rectas. Em coordenadas cartesianas, a caracterização de rectas é feita habitualmente através da sua equação reduzida:

$$y = mx + b \quad (5.10)$$

Onde  $m$  representa o declive e  $b$  a ordenada na origem. No entanto, os valores que  $m$  e  $b$  podem tomar são ilimitados, pelo que a sua utilização como parâmetros da transformada de

Hough complicaria a sua implementação. No caso particular deste projecto, pretende-se detectar o marcador independentemente da sua posição e orientação na imagem. Desta forma, as rectas que aproximam os lados do marcador podem ter diversas direcções, pelo que não existe uma maneira de as caracterizar, através do seu declive e ordenada na origem, que mantenha estes parâmetros com valores limitados.

Uma solução para este problema passa por definir as rectas em coordenadas polares. Neste sistema, uma linha recta é representada pela equação:

$$\rho = x \cos \theta + y \sin \theta \quad (5.11)$$

Onde  $\rho$  corresponde à distância da recta à origem e  $\theta$  ao ângulo que o vector normal faz com o eixo horizontal. Utilizando os eixos coordenados anteriormente definidos para uma *frame*, a equação 5.11 fica:

$$\rho = u \cos \theta + v \sin \theta \quad (5.12)$$

Cada ponto no plano da imagem corresponderá a uma curva no espaço da transformação, conforme se ilustra na figura 5.8. A transformação de pontos colineares resultará, no espaço da transformada, a um conjunto de curvas que se intersectam num ponto, cujas coordenadas  $(\theta_r, \rho_r)$  correspondem aos parâmetros da recta à qual pertencem no plano da imagem.

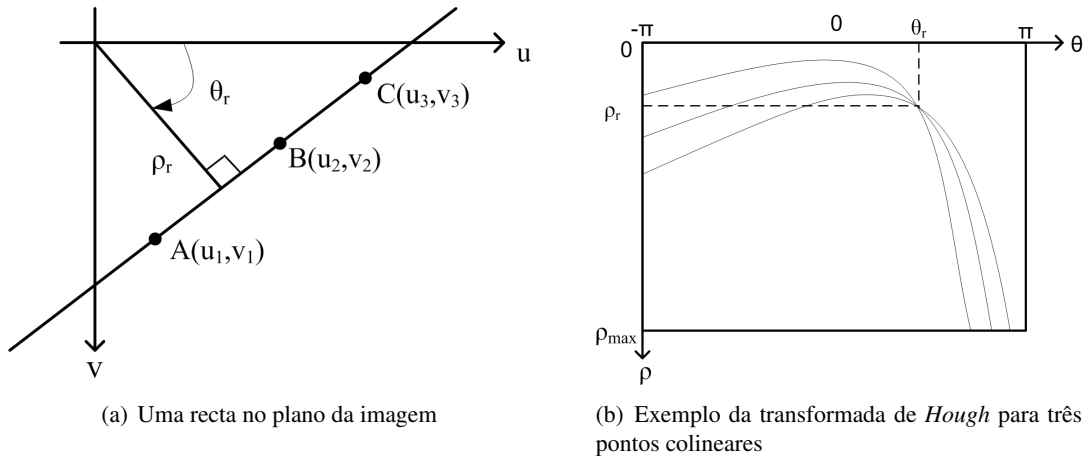


Figura 5.8: Uma recta no plano da imagem e exemplo da transformada de *Hough* para três dos seus pontos

A implementação em *software* da transformada de *Hough* obriga à discretização do espaço da transformada, que passa a ser representado por uma matriz, *MHough*, cujas dimensões dependem da resolução da discretização de  $\rho$  e  $\theta$ , respectivamente  $\Delta\rho$  e  $\Delta\theta$ . Uma vez que se pretende a detecção de rectas em toda a imagem e segundo todas as direcções possíveis, é necessário considerar, para  $\theta$ , os valores compreendidos no intervalo  $[-\pi; \pi]$ , e para  $\rho$  a gama  $[0; \rho_{max}]$ , onde  $\rho_{max}$  corresponde ao valor máximo da distância de uma recta à origem, ou seja, o comprimento da diagonal da imagem. A escolha da resolução da transformada é de grande importância devido

à sua influência no desempenho do algoritmo e será apresentada na secção 6.1. Se por um lado a utilização de intervalos de discretização baixos permite melhor resolução na estimação das rectas, o tempo de processamento da transformada é agravado e aumenta linearmente com o número de intervalos de discretização.

O algoritmo da transformada de *Hough* consiste em percorrer a imagem original e, para cada ponto de orla detectado, incrementar as posições da matriz *MHough* que melhor aproximam a curva da transformada que lhe está associada. Para tal, é calculado o  $\rho$  relativo a cada valor discreto da gama de  $\theta$ , através da equação 5.12. O conjunto de pares  $(\rho, \theta)$  obtidos estão relacionados com os índices da matriz que devem ser incrementados. No final do processo, a matriz resultante apresentará picos, ou valores máximos locais, nas posições cujos índices correspondem aos parâmetros das possíveis rectas existentes na imagem original. Na figura 5.9 está representada o resultado da transformação de uma *frame* captada na presença de um marcador. As zonas mais escuras correspondem a posições da matriz com valores elevados e que indiciam a presença dos segmentos de recta do marcador na imagem original.

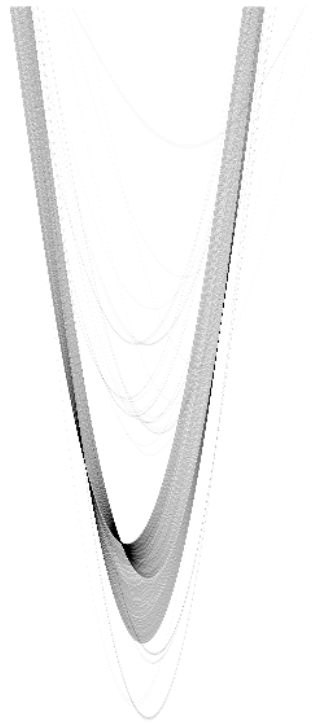


Figura 5.9: Exemplo do resultado da transformada de *Hough*

Após a transformação de todos os pontos de orla, é necessário verificar a existência de segmentos de recta semelhantes aos de um marcador e, em caso afirmativo, obter os seus parâmetros. O algoritmo de pesquisa implementado está representado na figura 5.10.

Inicialmente pesquisa-se o máximo global de *MHough*, encontrado na posição  $(j_{max}, i_{max})$ . O valor desse máximo,  $MHough[j_{max}, i_{max}]$ , corresponde ao número de pontos que a recta associada possui no espaço cartesiano da imagem [53], pelo que é possível verificar se esse valor se encontra dentro da gama  $[minPointsOnLine; maxPointsOnLine]$ . Os limites deste intervalo devem

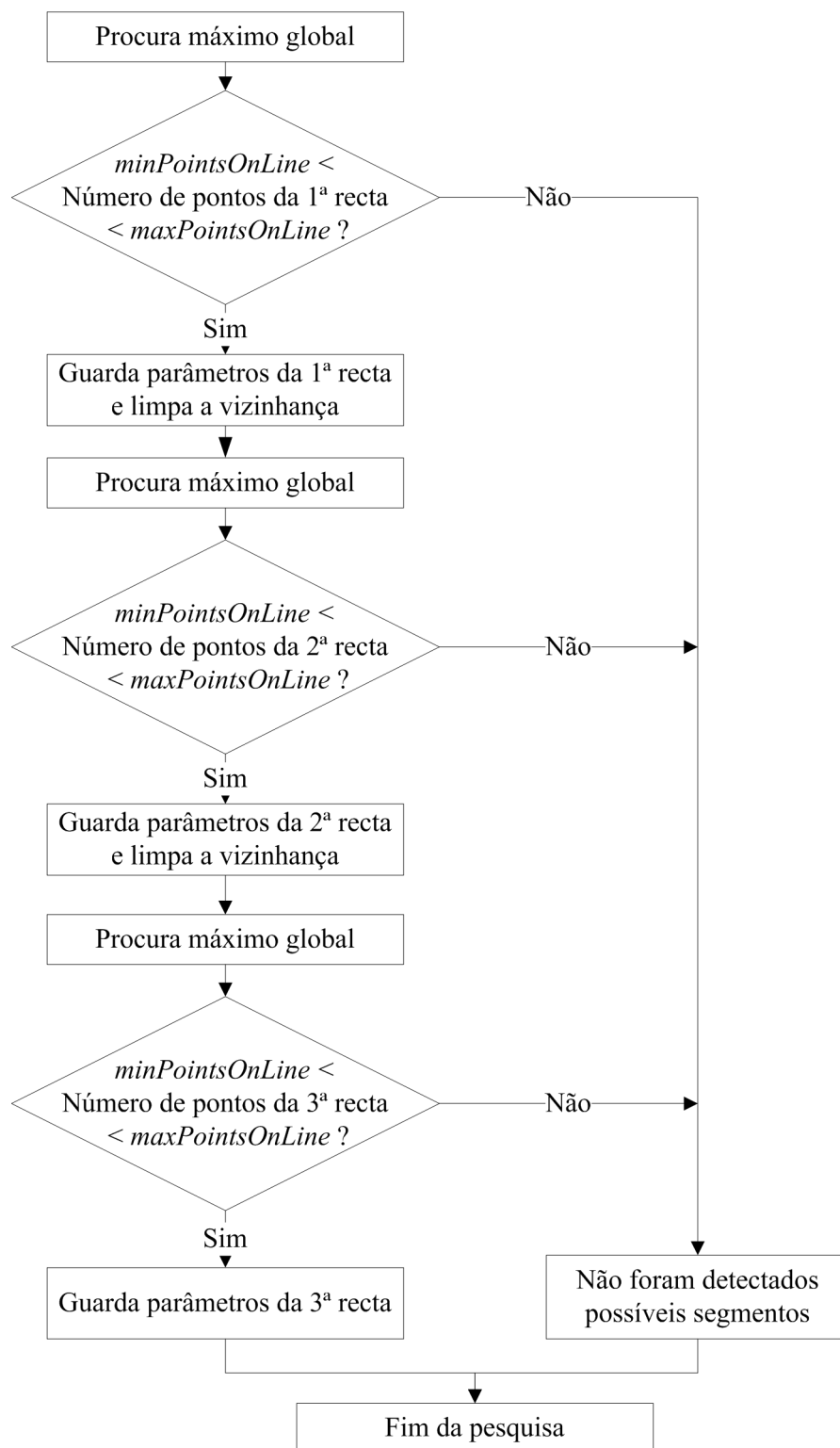


Figura 5.10: Algoritmo de pesquisa dos segmentos de recta do marcador

ser ajustados para uma gama bastante alargada de modo a permitir a detecção de segmentos de recta corrompidos ou parcialmente danificados. Este teste inicial permite despistar falsos positivos numa fase antecipada do algoritmo. Caso a recta detectada verifique esta condição inicial, o algoritmo guarda os índices desse máximo e calcula os parâmetros da recta associada através de uma média ponderada dos vizinhos:

$$\rho_a = \frac{MHough[j_{max}-1, i_{max}] * \rho[j_{max}-1] + MHough[j_{max}, i_{max}] * \rho[j_{max}] + MHough[j_{max}+1, i_{max}] * \rho[j_{max}+1]}{MHough[j_{max}-1, i_{max}] + MHough[j_{max}, i_{max}] + MHough[j_{max}+1, i_{max}]} \quad (5.13)$$

$$\theta_a = \frac{MHough[j_{max}, i_{max}-1] * \theta[i_{max}-1] + MHough[j_{max}, i_{max}] * \theta[i_{max}] + MHough[j_{max}, i_{max}+1] * \theta[i_{max}+1]}{MHough[j_{max}, i_{max}-1] + MHough[j_{max}, i_{max}] + MHough[j_{max}, i_{max}+1]} \quad (5.14)$$

Onde  $\rho[j]$  e  $\theta[i]$  representam respectivamente os valores de  $\rho$  e  $\theta$  associados ao índice  $(j, i)$  da matriz *MHough*. Com esta consulta dos vizinhos pretende-se minimizar os efeitos da discretização do espaço da transformada e obter uma estimativa mais precisa dos parâmetros da recta. Ao mesmo tempo, a consulta dos valores de apenas dois pontos adjacentes mantém o algoritmo simples e rápido.

Cada segmento de recta é definido internamente por dois pontos, o que permite caracterizar rectas em todas as direcções e com parâmetros que não assumam valores ilimitados, bem como utilizar uma função de visualização das mesmas na imagem capturada. Assim, a partir dos dados em coordenadas polares, as coordenadas no referencial da imagem dos dois pontos necessários podem ser obtidas, para rectas com declive “próximo da horizontal”, através de

$$P_1(x1, y1) = \left(0, \frac{\rho_a}{\sin(\theta_a)}\right) \quad (5.15)$$

$$P_2(x2, y2) = \left(ImgWidth - 1, \frac{\rho_a - (ImgWidth - 1) \cos \theta_a}{\sin \theta_a}\right) \quad (5.16)$$

e, para rectas “próximas da vertical”, através de

$$P_1(x1, y1) = \left(\frac{\rho_a}{\cos \theta_a}, 0\right) \quad (5.17)$$

$$P_2(x2, y2) = \left(-(ImgHeight - 1) \tan \theta_a + \frac{\rho_a}{\cos \theta_a}, ImgHeight - 1\right) \quad (5.18)$$

Para se garantir que a próxima pesquisa em *MHough* retorne um máximo local relativo a uma nova recta, e não a um pico elevado próximo da recta já obtida, é necessário limpar uma vizinhança da posição da matriz *MHough* processada. Como se sabe que os segmentos de recta do marcador têm direcções que distam entre si de pelo menos o menor ângulo interno do mesmo, é possível definir um intervalo em  $\theta$ ,  $[\theta_a - \theta_{clear}; \theta_a + \theta_{clear}]$  para o qual se pode

anular o valor da matriz  $MHough$ . O valor de  $thetaRangeClear$  deve ser ligeiramente inferior ao menor ângulo interno do marcador triangular, mas próximo deste.

Este procedimento de pesquisa de rectas na matriz de transformação repete-se até que sejam identificadas três candidatas possíveis, excepto se numa das iterações o comprimento da recta não pertencer ao intervalo de aceitação, caso para o qual o algoritmo termina a pesquisa. Na figura 5.11 apresenta-se um exemplo da detecção de rectas a partir dos pontos de orla identificados.

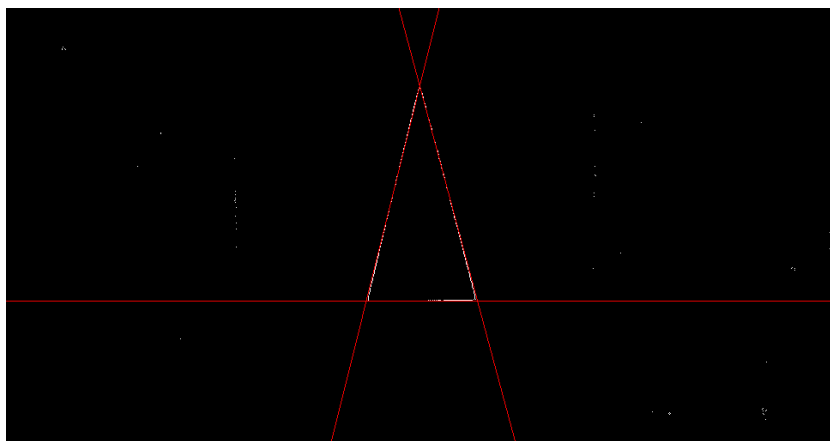


Figura 5.11: Detecção das rectas da imagem a partir dos pontos de orla

## 5.3 Calibração da câmara

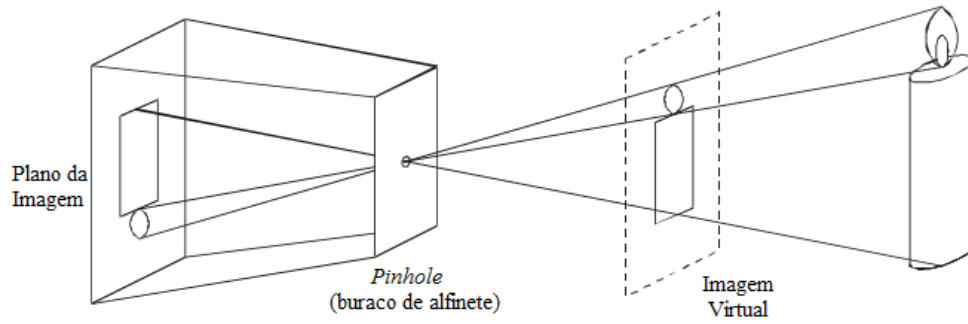
Para além da correcta detecção do marcador na imagem, torna-se necessário contextualizar esta informação no referencial do robô. A calibração da câmara tem por objectivo a determinação dos parâmetros do modelo matemático do sistema de visão, o que permite conhecer a posição no mundo de um dado pixel da imagem e, de modo inverso, o local no plano da imagem onde será mapeado um ponto do mundo.

Normalmente, os parâmetros da câmara são divididos em dois grupos: os parâmetros intrínsecos e os parâmetros extrínsecos. Os do primeiro grupo representam a distância focal, a altura e largura dos pixels do CCD e o centro da imagem, pelo que não variam com o posicionamento da câmara no mundo. Já os parâmetros extrínsecos, independentes dos intrínsecos, caracterizam a posição e rotação da câmara em relação ao referencial do robô [10].

### 5.3.1 Modelo da câmara

Nas câmaras convencionais, é frequente o uso de um modelo de projecção em perspectiva, comumente conhecido como modelo *pinhole* (ver figura 5.12). Segundo este modelo, uma câmara é vista como uma caixa à prova de luz onde existe um pequeno orifício (*pinhole*), idealmente considerado como apenas um ponto. A luz oriunda de todo o espaço tridimensional à frente do orifício passa por este e é projectada no plano da imagem, daí o nome de projecção perspectiva.



Figura 5.12: Projecção perspectiva no modelo *pinhole* [9]

Para obter o modelo *pinhole* da câmara é necessário, antes de mais, definir os diversos sistemas de coordenadas, que foram especificados conforme se ilustra na figura 5.13.

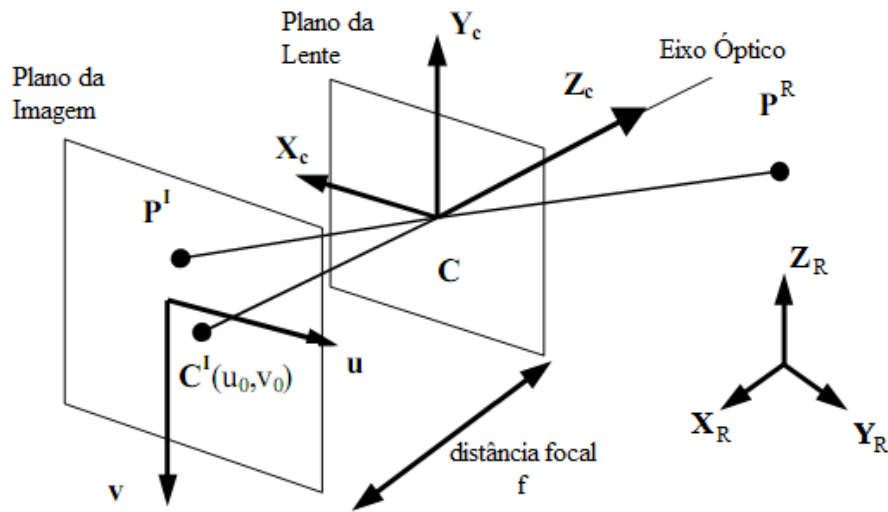


Figura 5.13: Sistemas coordenados especificados [10]

Todos os sistemas coordenados são cartesianos. No referencial do robô,  $OX_RY_RZ_R$ , um ponto genérico possui coordenadas  $P^R = (x, y, z)$ . A posição da câmara no sistema coordenado do robô é definida pelo ponto  $C = (x_{c0}, y_{c0}, z_{c0})$ , que é igualmente a origem do referencial da câmara,  $OX_CY_CZ_C$ . A projecção de um ponto no plano da imagem é mapeada num pixel de coordenadas  $P^I = (u, v)$ . O pixel central de uma *frame* tem coordenadas  $C^I = (u_0, v_0)$ . A distância focal,  $f$ , corresponde à distância entre o plano da lente e o plano da imagem.

Da figura 5.13, podemos concluir que um pixel genérico  $P^I$  corresponde à projecção de um ponto  $P^R$  do mundo no plano da imagem, podendo a sua relação ser modelada através da seguinte equação:

$$P^I = H \cdot P^R \quad (5.19)$$

A matriz  $H$ , matriz de projecção da câmara, pode ser decomposta numa série de transformações lineares entre referenciais.

A transformação das coordenadas no referencial do robô de um ponto em coordenadas no referencial da câmara equivale à seguinte matriz de transformação homogênea:

$$H_C^R = \begin{bmatrix} R_C^R & T_C^R \\ 0 & 1 \end{bmatrix} \quad (5.20)$$

$T_C^R = [t_x \ t_y \ t_z]^T$  representa o vector de translação entre os dois sistemas coordenados. A rotação entre os dois referenciais é modelada pela matriz  $R_C^R$ , que pode ser caracterizada pelos três ângulos de *Euler*,  $(\phi_x, \phi_y, \phi_z)$ :

$$R_C^R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c_x c_y & s_x c_z + c_x s_y s_z & s_x s_z - c_x s_y c_z \\ -s_x c_y & c_x c_z - s_x s_y s_z & c_x s_z + s_x s_y c_z \\ s_y & -c_y s_z & c_y c_z \end{bmatrix} \quad (5.21)$$

Onde  $c_x = \cos \phi_x$ ,  $s_x = \sin \phi_x$  e de forma análoga para os restantes termos associados com  $\phi_y$  e  $\phi_z$ .

Neste momento, encontra-se descrita a relação entre as coordenadas de um dado ponto  $P$  no referencial do robô,  $OX_R Y_R Z_R$ , e as coordenadas do mesmo ponto no referencial da câmara,  $OX_C Y_C Z_C$ .

De seguida, o ponto no plano da imagem é obtido através de uma projecção perspectiva. Esta operação, que converte as coordenadas tridimensionais do ponto no referencial da câmara em coordenadas a duas dimensões, pode ser representada, em termos matriciais, por:

$$H_F^C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \quad (5.22)$$

Como já definido,  $f$  é a distância focal, isto é, a distância entre o plano da lente e o plano da imagem.

Finalmente, as coordenadas no plano da imagem são obtidas depois de se aplicar um factor de escala para conversão de unidades de distância em pixels e uma translação de acordo com as coordenadas do pixel central da imagem. Esta transformação pode ser representada através da seguinte matriz:

$$H_I^F = \begin{bmatrix} \frac{-1}{du} & 0 & u_0 \\ 0 & \frac{-1}{dv} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.23)$$

Os factores  $du$  e  $dv$  representam, respectivamente, a resolução horizontal e vertical do CCD. De salientar que os elementos de  $H_I^F$  associados a  $du$  e  $dv$  têm sinal negativo, uma vez que os eixos  $u$  e  $v$  do plano da imagem têm sentido contrário aos eixos  $x$  e  $y$  do referencial da câmara.

Desta forma, a transformação de coordenadas no referencial do robô para o plano da imagem, apresentada na equação 5.19, pode ser reescrita do seguinte modo:

$$P^I = H_I^F \cdot H_F^C \cdot H_C^R \cdot P^R \quad (5.24)$$

Esta relação de transformação foi implementada na função *xyz2uv*, que a partir das coordenadas no referencial do robô de um ponto retorna as coordenadas correspondentes no plano da imagem, em pixels.

De salientar que o sistema de equações definido em 5.24 não é invertível, isto é, partindo das coordenadas de um pixel no plano da imagem obtemos uma infinidade de pontos no mundo cuja projecção perspectiva é a mesma. No entanto, como todos esses pontos pertencem a uma mesma recta, é possível determinar as coordenadas de um ponto no referencial do robô associado a um dado pixel, definindo um dos parâmetros em falta. Esta relação foi implementada na função *uvz2xy*, que recebe como parâmetros de entradas as coordenadas  $(u, v)$  de um pixel no plano da imagem e o valor de cota no referencial do robô,  $z$ , para o qual se pretende o cálculo das coordenadas  $x$  e  $y$  do ponto.

### 5.3.2 Calibração dos Parâmetros de Transformação

Para que as funções que implementam as relações de transformação do modelo da câmara (mencionadas na secção 5.3.1) possam funcionar, é necessário determinar os parâmetros de calibração da câmara. Vários métodos existentes de estimação destes parâmetros baseiam, habitualmente, o seu princípio de funcionamento no conhecimento rigoroso das coordenadas mundo e das respectivas coordenadas no plano da imagem de um dado conjunto de pontos [54].

O sistema de equações 5.24 possui 11 parâmetros que necessitam de ser determinados: 6 correspondentes aos parâmetros extrínsecos da câmara, associados à rotação e translação entre os referenciais da câmara e do robô ( $\phi_x, \phi_y, \phi_z, t_x, t_y$  e  $t_z$ ), e os restantes 5 aos parâmetros intrínsecos, isto é,  $f, du, dv, u_0$  e  $v_0$ . Sendo assim, a utilização de um algoritmo de correspondência de pontos implicaria a obtenção de, pelo menos, 6 pontos cujas coordenadas no referencial do robô e no plano da imagem fossem conhecidas.

Devido ao sucesso alcançado num trabalho recente com uma câmara semelhante, recorreu-se a um método de calibração mais simples, baseado no conhecimento de apenas dois pontos [39]. Para isso foram realizadas algumas simplificações ao modelo anteriormente apresentado, que passo a citar:

- a posição da câmara no referencial do robô pode ser medida após a conclusão da sua instalação, pelo que a translação do referencial da câmara em relação ao sistema de coordenadas do robô,  $T_C^R$ , passa a ser conhecida;
- os valores da resolução horizontal e vertical do CCD,  $du$  e  $dv$ , foram obtidos a partir da documentação técnica do fabricante;

- as coordenadas do centro da imagem correspondem ao pixel central, ou seja,  $u_0 = \frac{ImgWidth}{2}$  e  $v_0 = \frac{ImgHeight}{2}$ .

Posto isto, resta determinar a rotação da câmara em relação ao referencial do robô e a distância focal equivalente. A rotação entre  $OX_RY_RZ_R$  e  $OX_CY_CZ_C$  pode ser definida pela composição das rotações simples segundo cada um dos eixos coordenados,  $Rot_{\theta_x}$ ,  $Rot_{\theta_y}$  e  $Rot_{\theta_z}$ . Os quatro parâmetros ainda desconhecidos (os três ângulos de rotação  $\theta_x$ ,  $\theta_y$  e  $\theta_z$ , bem como a distância focal,  $f$ ), serão estimados com base no conhecimento de apenas dois pontos de calibração, segundo um algoritmo que se passa a descrever.

O primeiro ponto de calibração corresponde ao pixel central da imagem,  $P_c^I = (u_0, v_0)$ , cujas coordenadas no referencial do robô,  $(x_{P_c}, y_{P_c}, z_{P_c})$ , são obtidas por medição. Analisando a configuração dos referenciais do sistema de visão, ilustrada na figura 5.14, conclui-se que os valores dos ângulos de rotação segundo os eixos  $x$  e  $y$  podem ser obtidos através de:

$$\theta_x = \arctan2\left(\frac{y_{c0} - y_{P_c}}{\sqrt{(z_{c0} - z_{P_c})^2 + (x_{c0} - x_{P_c})^2}}\right) \quad (5.25)$$

$$\theta_y = \arctan2\left(\frac{x_{P_c} - x_{c0}}{z_{P_c} - z_{c0}}\right) \quad (5.26)$$

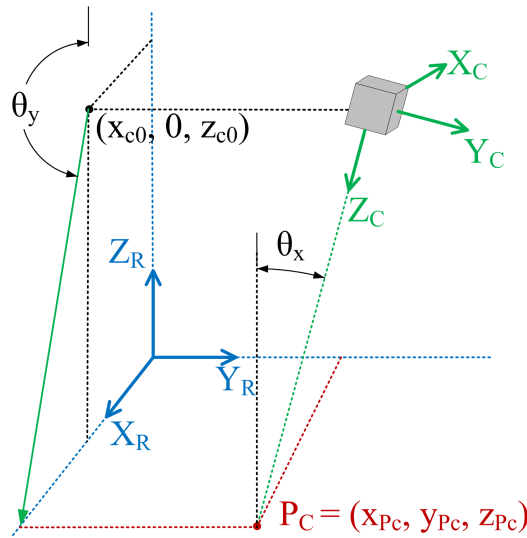


Figura 5.14: Rotação do referencial da câmara em relação ao sistema coordenado do robô

A determinação do ângulo de rotação  $\theta_z$  recorre ao segundo ponto de calibração,  $P_a$ , um qualquer ponto da *frame* cuja posição no sistema de coordenadas do robô seja conhecida. Inicializa-se  $\theta_z$  com valor nulo e atribui-se à distância focal um valor aleatório, uma vez que este último parâmetro não é ainda conhecido e, como actua apenas como um factor de escala, não afectará o resultado do cálculo. Aplicando as coordenadas no referencial do robô de  $P_a$  à função  $xyz2uv$  obtemos os valores  $(u'_{P_a}, v'_{P_a})$ . A diferença de ângulo entre este ponto e as coordenadas reais no plano,  $(u_{P_a}, v_{P_a})$ , corresponde exactamente a  $\theta_z$ .

Finalmente, resta estimar o valor da distância focal equivalente. Neste passo são usados todos os parâmetros já determinados, incluindo  $\theta_z$ , pelo que as coordenadas de  $P_a$  no plano da imagem devolvidas pela função  $xyz2uv$  necessitam de ser recalculadas, obtendo-se  $(u''_{P_a}, v''_{P_a})$ . Uma vez que a distância focal não foi ainda ajustada, todos os pares  $(u, v)$  retornados pela função  $xyz2uv$ , apesar de apresentarem a mesma direcção, possuem uma norma diferente da real. Este facto pode ser corrigido comparando o valor calculado com as verdadeiras coordenadas de  $P_a$  no plano da imagem,  $(u_{P_a}, v_{P_a})$ , que são conhecidas. A estimativa da distância focal pode por isso ser determinada da seguinte forma:

$$f = \frac{\|(u_{P_a}, v_{P_a})\|}{\|(u''_{P_a}, v''_{P_a})\|} \quad (5.27)$$

Desta forma, os parâmetros de calibração da câmara ficam totalmente determinados, permitindo a transformação das coordenadas no plano da imagem em coordenadas no referencial do robô.

## 5.4 Identificação do marcador

Após a detecção de três rectas candidatas resta testar a presença de um marcador e, em caso afirmativo, calcular o seu centro e orientação. O algoritmo implementado para identificação e validação do marcador está representado na figura 5.15.

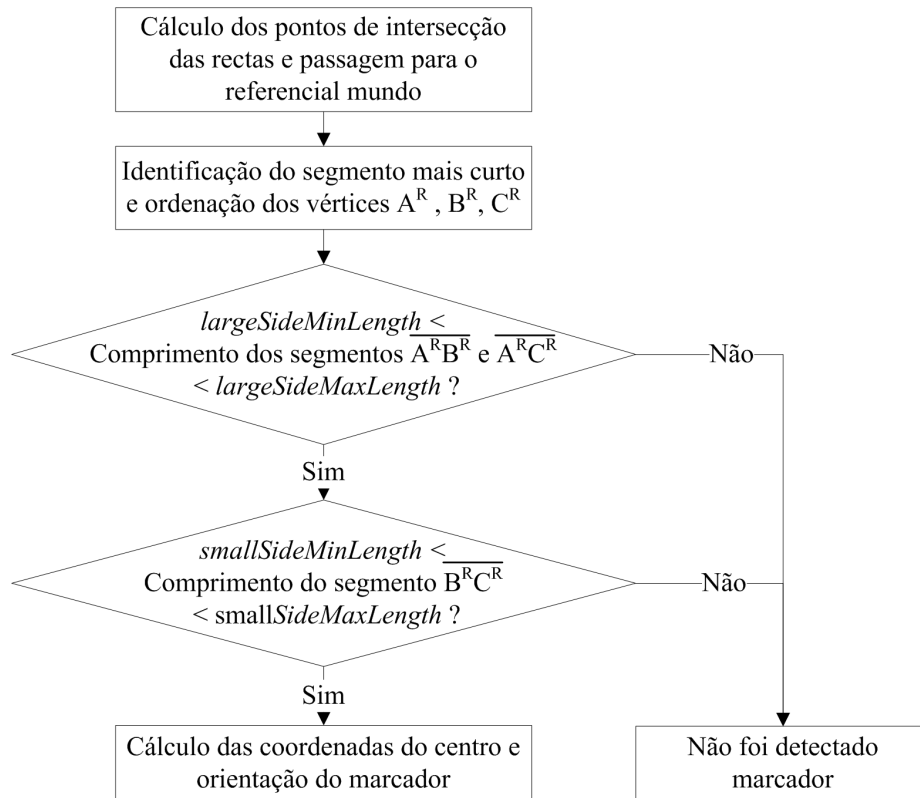


Figura 5.15: Algoritmo de identificação e validação do marcador

Como o marcador é plenamente caracterizado pelos seus três vértices, são calculadas as coordenadas no plano da imagem dos pontos de intersecção das rectas candidatas identificadas,  $A^I$ ,  $B^I$  e  $C^I$  (ver Anexo A). Os pontos obtidos são transformados para o referencial do robô, utilizando a função  $uvz2xy$  (mencionada na secção 5.3.1), obtendo-se  $A^R$ ,  $B^R$  e  $C^R$ . De seguida, através da distância entre estes três pontos, é identificado o segmento mais curto e reorganizados os vértices de modo a que  $\overline{B^R C^R}$  corresponda ao segmento mais curto e o vértice  $A_R$  ao indicador da orientação do marcador.

O próximo passo consiste em verificar se a forma detectada corresponde a um triângulo com as dimensões esperadas. Para isso, os comprimentos dos segmentos mais largos,  $\overline{A^R B^R}$  e  $\overline{A^R C^R}$ , bem como do segmento mais curto,  $\overline{B^R C^R}$ , são comparados com dois intervalos, respectivamente  $[largeSideMinLength; largeSideMaxLength]$  e  $[smallSideMinLength; smallSideMaxLength]$ . Estes intervalos são centrados nas dimensões reais conhecidas do marcador e as suas larguras comportam-se como tolerâncias em relação às mesmas. Estas tolerâncias devem ser ajustadas para valores tais que permitam a identificação de marcadores parcialmente degradados, mas, ao mesmo tempo, suficientemente restritos para despistar falsas detecções e garantir a robustez do sistema.

Por fim, na presença de um triângulo válido, resta determinar a posição e a orientação do marcador no sistema coordenado do robô. Uma vez que os pontos  $A^R$ ,  $B^R$  e  $C^R$  já definidos em relação ao referencial do robô, o centro é calculado como o “centro de massa” desses pontos, ou seja,  $\frac{A^R + B^R + C^R}{3}$ . A orientação do marcador corresponde à orientação do vector  $\overrightarrow{D^R A^R}$ , em que  $D^R$  é o ponto intermédio do segmento  $\overline{B^R C^R}$  e  $A^R$  o vértice indicador.

Na figura 5.16 encontra-se o resultado do processo de identificação de um marcador, com indicação dos pontos de interesse, bem como do centro e orientação (a azul) calculados. Conforme é possível visualizar, o algoritmo identifica correctamente o marcador a partir dos pontos de orla detectados.

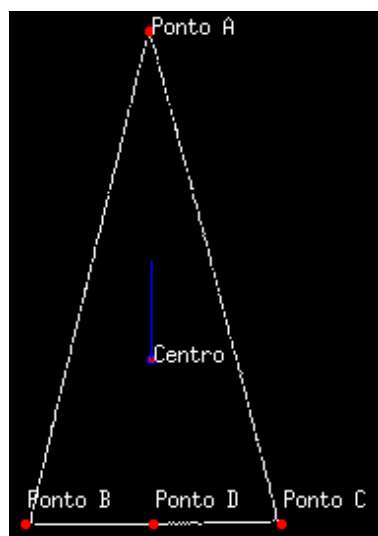


Figura 5.16: Pormenor da identificação dos vértices, centro e orientação do marcador

## 5.5 Cálculo da incerteza

Como em qualquer processo de identificação, é vantajoso ter uma estimativa da qualidade da medida obtida pelo algoritmo de detecção dos marcadores visuais. O critério utilizado baseia a medida da incerteza em dois factores principais: a qualidade da detecção das rectas (associada à diferença entre o número de pontos de cada recta identificada e o número de pontos esperado) e o grau de semelhança entre o triângulo obtido e as dimensões reais do marcador, que são conhecidas. Sendo assim,

$$uncertainty = k_1 \left| \frac{\Sigma IdealPoints - \Sigma DetectedPoints}{\Sigma IdealPoints} \right| + k_2 \left| \frac{IdealPerimeter - DetectedPerimeter}{IdealPerimeter} \right| \quad (5.28)$$

Onde  $k_1$  e  $k_2$  representam os pesos dados a cada componente,  $\Sigma IdealPoints$  o número total de pontos de um marcador ideal,  $\Sigma DetectedPoints$  o número total de pontos das três rectas detectadas (obtidos da matriz  $MHough$ ),  $IdealPerimeter$  o perímetro real do marcador conhecido e  $DetectedPerimeter$  a soma dos comprimentos dos segmentos  $\overline{A^R B^R}$ ,  $\overline{A^R C^R}$  e  $\overline{B^R C^R}$ .

## 5.6 Identificação do código do marcador

Adicionalmente ao algoritmo de detecção de marcadores, foi desenvolvida uma metodologia de codificação dos marcadores que permite a sua distinção unívoca. Esta característica complementar permite que, após a identificação de um marcador, a pose do veículo no referencial global fique totalmente definida, independentemente da estimativa actual (ver secção 4.2).

O código adoptado consiste de 6 zonas circulares colocadas no interior do marcador, conforme se exemplifica na figura 5.17. Estas zonas comportam-se como um código binário: atribuindo-se o valor lógico 0 para a ausência e o valor 1 para a presença desta “mancha” sobre a posição esperada no marcador. Associando a cada posição dos pontos do código um peso diferente, é possível gerar até  $2^6 = 64$  combinações.

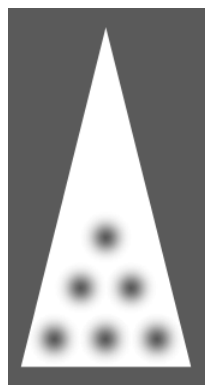


Figura 5.17: Exemplo de marcador com as zonas de código

Estas manchas são projectadas de modo a que a sua zona central tenha elevado contraste com o fundo do marcador. Deste modo, é possível atribuir a cada bit do código o seu valor por simplesmente avaliar a tonalidade da zona correspondente. No caso de um marcador com fundo branco, os bit's cuja região apresente tonalidade escura devem ser colocados a 1 e os restantes com valor 0.

A transição entre a tonalidade do fundo do marcador e as manchas que constituem o código identificador é muito suave. Assim, não são detectados pontos de orlas associados a estas zonas interiores, pelo que a presença desta codificação é imperceptível ao algoritmo que calcula o centro e a orientação do marcador, descrito anteriormente.

Na figura 5.18 apresenta-se o algoritmo de identificação do código do marcador.

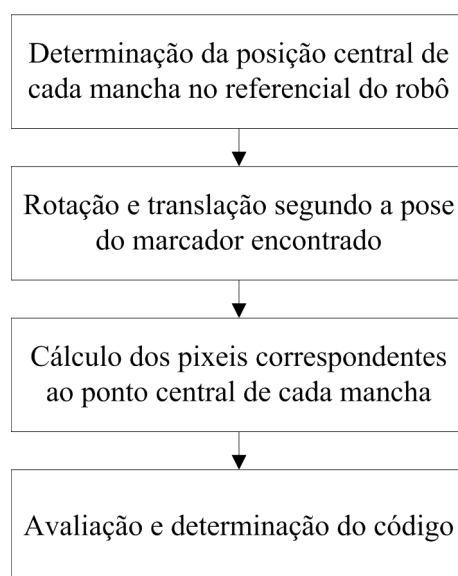


Figura 5.18: Algoritmo de identificação do código do marcador

A posição central de cada mancha em relação aos limites do marcador é conhecida na fase de projecto. Assume-se que cada uma dessas coordenadas,  $(CodePoint.x_0, CodePoint.y_0)$ , são medidas em relação ao referencial do robô, com o marcador centrado na origem e orientado segundo o eixo  $x$ . Assim, quando é detectado um marcador no mundo, as coordenadas dos centros das manchas são directamente determinadas. Para isso, basta aplicar a cada ponto  $(CodePoint.x_0, CodePoint.y_0)$  uma rotação segundo  $DetectedMark.orientation_w$ , seguida de uma translação de  $(DetectedMark.x_w, DetectedMark.y_w)$ , isto é, a orientação e as coordenadas do centro do marcador detectado.

De seguida, é necessário determinar a região da imagem onde será mapeada cada mancha e calcular o respectivo código. Para isso recorre-se à função  $xyz2uv$  (ver secção 5.3.1), que, recorde-se, devolve as coordenadas do pixel onde é mapeado um qualquer ponto do mundo. Aplicando a esta função as coordenadas mundo do centro de uma mancha, obtêm-se as respectivas coordenadas no plano da imagem, isto é,  $(CodePoint.u, CodePoint.v)$ . Nesta fase, a localização na imagem dos 6 pontos do código passam a ser conhecidas, faltando apenas testar o nível de cinzento desses



pontos e determinar o código do marcador. Na figura 5.19 exemplifica-se as regiões da imagem que o algoritmo utilizou para calcular o código do marcador.

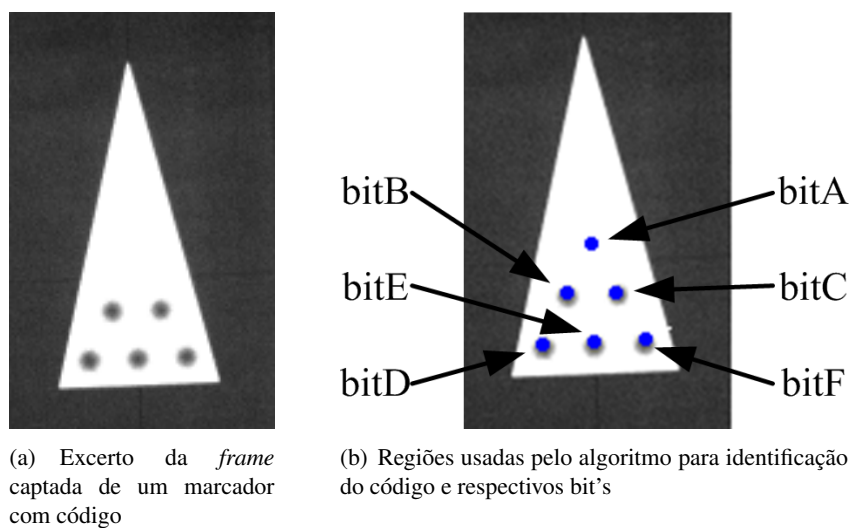


Figura 5.19: Código identificador do marcador

Para aumentar a imunidade ao ruído, avalia-se, para cada posição do código, não só o pixel central obtido, mas também os 4 vizinhos adjacentes. A média dos níveis de cinzento desses 5 pixels é comparada com um dado valor limite. Esta operação corresponde a um processo de binarização e permite obter imediatamente o valor de cada bit do código. Caso se esteja na presença de uma mancha, o nível de cinzento médio destes 5 pixels será baixo, uma vez que a mancha apresenta tonalidade escura. Neste caso, atribui-se o valor 1 ao bit do código. Contrariamente, a codificação nula de um bit é indiciada pelo elevado valor médio do nível de cinzento dos pixels em análise, uma vez que o fundo do marcador é claro. De referir que esta lógica se inverteria caso fosse usado um marcador de fundo escuro.

O valor do código, compreendido entre 0 e 63, é calculado da seguinte forma:

$$DetectedMark.code = 1 * bitA + 2 * bitB + 4 * bitC + 8 * bitD + 16 * bitE + 32 * bitF \quad (5.29)$$

Esta técnica de identificação unívoca do marcador pode suscitar problemas, caso algumas das manchas sejam degradadas ou ocultas. Porém, nestas circunstâncias prevê-se que a incerteza associada à detecção do marcador também aumente, pelo que a utilização de limiares de aceitação restritos poderá minimizar os seus efeitos.

Outra abordagem para melhorar a robustez da identificação do código do marcador passa por utilizar apenas uma parte do total de códigos disponíveis. Neste caso, dos 63 códigos não nulos permitidos, optou-se por utilizar apenas os 4 representados na figura 5.20.

Para este conjunto, a atribuição de um valor incorrecto a um dos bit's do código resultará numa sequência inválida, o que conduzirá à rejeição da amostra.

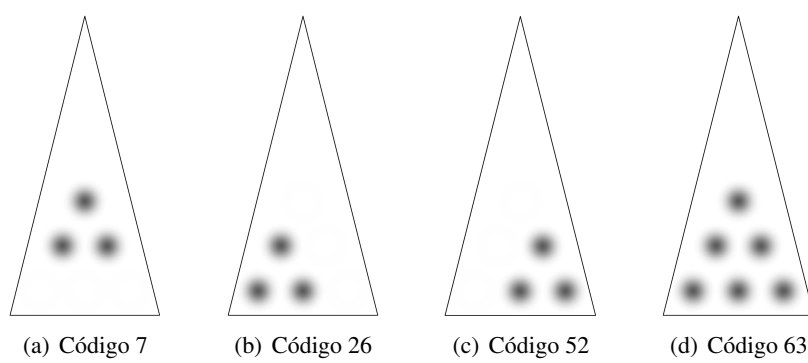


Figura 5.20: Exemplo dos códigos usados

## Capítulo 6

# Testes e resultados experimentais

Neste capítulo serão apresentados os resultados experimentais obtidos durante a realização de testes aos variados sistemas.

### 6.1 Testes ao sistema de detecção de marcadores visuais

A fim de testar a eficiência dos algoritmos de detecção de marcadores visuais desenvolvidos, é necessário avaliar a sua resposta quando o marcador aparece em diferentes posições na imagem. Assim, foram tiradas e guardadas *frames* de teste com marcadores colocados em 3 diferentes posições na imagem e segundo 8 sentidos direccionais distintos, espaçados entre si de  $45^\circ$ , totalizando 24 amostras. As poses ( $x$ ,  $y$ ,  $\theta$ ) reais do marcador para cada imagem de teste foram registadas, de modo a se poder quantificar e comparar o erro da estimativa dada pelo sistema de detecção.

Conforme já referido, o tempo de processamento da transformada de *Hough* está intimamente relacionado com a resolução de discretização ( $\Delta\rho$ ,  $\Delta\theta$ ) escolhida. Assim, para além de detectar o marcador de forma suficientemente precisa e robusta, torna-se necessário garantir que a configuração da transformada a utilizar pelo algoritmo satisfaça os requisitos temporais definidos. Analisou-se a resposta do algoritmo às 24 amostras para diferentes configurações da resolução da transformada de *Hough*. Uma síntese dos resultados encontram-se na tabela 6.1. O tempo de processamento do algoritmo para cada par ( $\Delta\rho$ ,  $\Delta\theta$ ), testado com a transformação de 610 pontos da imagem, é apresentado na figura 6.1.

A partir dos dados obtidos é possível retirar algumas conclusões. Conforme se constata na tabela 6.1 e na figura 6.2, o erro da estimativa da pose do marcador permanece praticamente constante, principalmente para os primeiros pares de resoluções ( $\Delta\rho$ ,  $\Delta\theta$ ), subindo ligeiramente com o aumento dos valores de resoluções utilizados. No entanto, deve-se ter em atenção que algumas amostras deixam de ser identificadas a partir do par número 13, o que impossibilita de todo a utilização destas configurações num sistema robusto. Além disso, verifica-se que a dispersão dos valores de erro é limitada, pelo que se pode assumir que a distorção em barril não é, nas condições deste projecto, um aspecto crítico.

Tabela 6.1: Resposta do algoritmo a 24 *frames* de teste para diferentes configurações da transformada de *Hough*, utilizando a resolução total de imagem

Par N.º	Resolução		Erro Médio			Erro Absoluto Médio			Erro Absoluto Máximo			Desvio Padrão do Erro			Número de Amostras Não Detectadas	Tempo de Processamento Médio (ms)
	$\Delta\rho$ (cm)	$\Delta\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)		
1	0,4	0,5	0,02	0,06	0,08	0,08	0,11	0,50	0,25	0,31	1,60	0,10	0,11	0,69	0	99
2	2,5	0,5	0,02	0,03	0,16	0,08	0,10	0,56	0,24	0,19	1,40	0,10	0,11	0,71	0	85
3	0,4	1	0,02	0,05	0,14	0,08	0,11	0,56	0,21	0,30	1,60	0,10	0,12	0,74	0	60
4	0,8	1	0,02	0,06	0,13	0,09	0,12	0,54	0,21	0,30	1,70	0,11	0,12	0,72	0	55
5	1,7	1	0,03	0,06	0,02	0,08	0,12	0,66	0,19	0,35	1,70	0,09	0,12	0,82	0	53
6	2,5	1	0,02	0,06	0,17	0,10	0,10	0,46	0,26	0,24	1,50	0,12	0,10	0,58	0	53
7	0,4	2	0,03	0,06	0,02	0,09	0,10	0,70	0,23	0,32	1,80	0,11	0,11	0,82	0	41
8	0,8	2	0,02	0,06	-0,10	0,08	0,10	0,79	0,23	0,29	1,70	0,10	0,11	0,93	0	38
9	1,7	2	0,02	0,07	0,04	0,09	0,12	0,80	0,22	0,26	1,90	0,11	0,12	0,96	0	38
10	2,5	2	0,00	0,07	0,12	0,09	0,12	0,75	0,28	0,28	2,20	0,12	0,12	0,96	0	37
11	0,4	3	0,03	0,06	0,24	0,10	0,13	0,83	0,25	0,25	1,20	0,12	0,14	0,86	0	35
12	0,8	3	0,01	0,07	0,15	0,08	0,12	0,79	0,24	0,27	1,10	0,10	0,13	0,83	0	33
13	1,7	3	0,01	0,06	0,17	0,09	0,12	0,94	0,28	0,32	2,10	0,12	0,13	1,01	1	32
14	2,5	3	0,01	0,08	-0,02	0,11	0,13	0,83	0,31	0,28	2,20	0,13	0,13	0,95	1	32
15	0,4	4	0,04	0,08	0,58	0,10	0,12	1,15	0,27	0,28	3,00	0,13	0,13	1,29	0	32
16	0,8	4	0,03	0,08	0,50	0,11	0,12	1,23	0,28	0,24	3,00	0,14	0,12	1,39	0	30
17	1,7	4	0,00	0,07	0,50	0,10	0,12	1,15	0,24	0,23	3,00	0,13	0,12	1,34	0	30
18	2,5	4	0,01	0,09	0,53	0,11	0,15	1,31	0,27	0,42	3,30	0,13	0,17	1,43	0	30
19	0,4	6	0,01	0,10	-0,05	0,15	0,15	0,87	0,61	0,47	3,20	0,22	0,16	1,18	1	30
20	2,5	6	0,01	0,12	-0,12	0,16	0,18	0,90	0,32	0,34	3,20	0,19	0,18	1,21	3	28

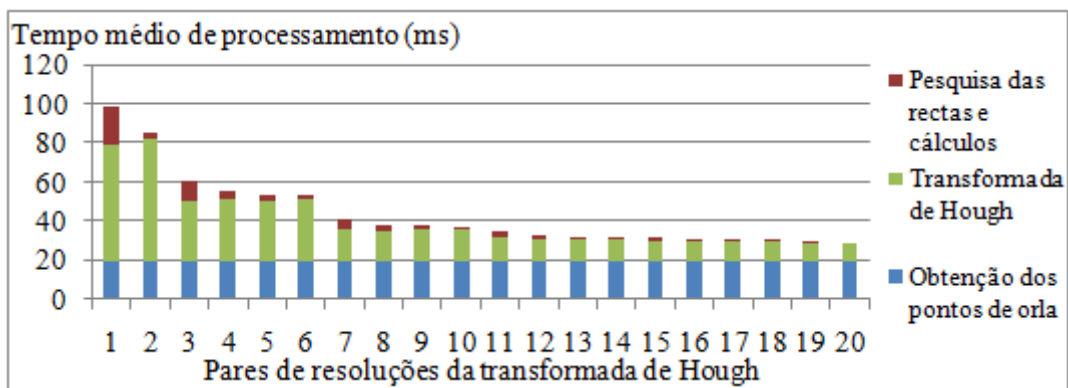


Figura 6.1: Tempo médio de processamento do algoritmo para diferentes configurações da transformada de *Hough*, utilizando a resolução total de imagem (610 pontos transformados)

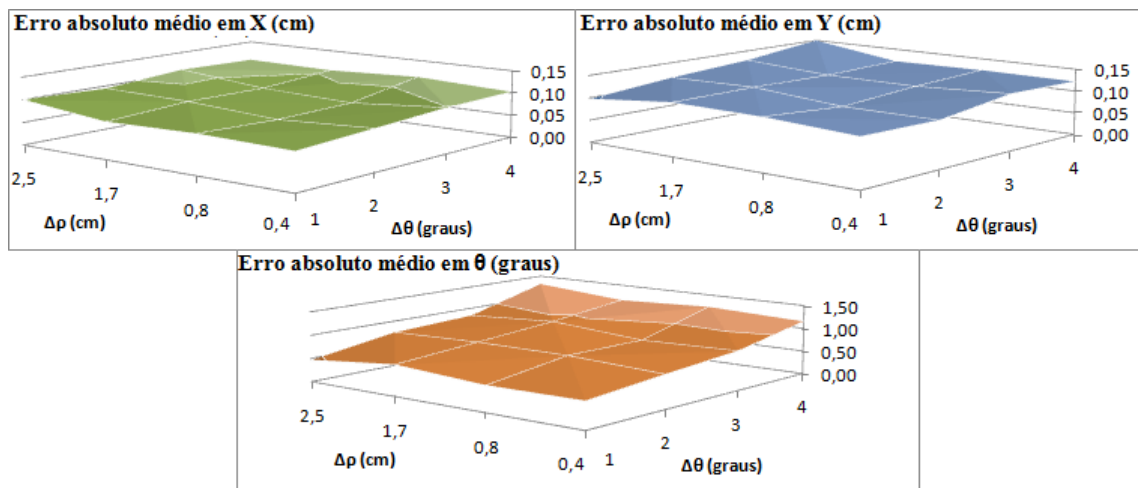


Figura 6.2: Evolução do erro da detecção do marcador visual com a resolução da transformada de *Hough*

Analisemos agora a figura 6.1. O tempo de processamento do algoritmo, factor restritivo neste sistema, é maioritariamente imposto pela duração da transformada de *Hough*. Como um ciclo de execução não pode exceder 66 ms, os pares 1 e 2 ficam imediatamente excluídos. Como seria de esperar, verifica-se que o tempo médio do algoritmo decresce com o aumento dos valores das resoluções. De salientar que o valor de  $\Delta\theta$  é especialmente crítico e exerce uma influência preponderante no desempenho do algoritmo.

Tomando em conta as conclusões anteriores, os pares de resoluções de maior interesse são aqueles com  $\Delta\theta = 1\text{ grau}$  e  $\Delta\theta = 2\text{ graus}$ , pelo que se optou por analisar os pares 4 e 9. Para estas configurações, o erro absoluto médio da estimativa fornecida pelo algoritmo foi de aproximadamente um milímetro, em  $x$  e  $y$ , e inferior a um grau para a orientação  $\theta$ .

Para as resoluções seleccionadas, verifica-se ainda que um parcela significativa do tempo de processamento do algoritmo está relacionada com as operações de gradiente e obtenção dos pontos de orla da imagem. Como estas operações são realizadas sobre todos os pixels da *frame*, a utilização de apenas uma fracção da resolução de imagem disponível ( $ImgWidth \times ImgHeight$ )

permite diminuir o número de pixels em análise e, conseqüentemente, o tempo de execução destas fases do algoritmo.

Assim, foram realizados testes semelhantes aos descritos anteriormente, mas agora utilizando apenas metade da resolução da imagem captada. Para isso, após a recepção de uma nova *frame*, realiza-se uma reamostragem obtendo-se uma matriz de pixels de dimensões  $(\frac{ImgWidth}{2} \times \frac{ImgHeight}{2})$ . Para reduzir o ruído, cada posição desta matriz é obtida pela média dos 4 pixels correspondentes da *frame* original:

$$F_{resample}[u, v] = \frac{F[2u, 2v] + F[2u + 1, 2v] + F[2u, 2v + 1] + F[2u + 1, 2v + 1]}{4} \quad (6.1)$$

Este procedimento não implica qualquer alteração das rotinas desenvolvidas: do ponto de vista do algoritmo apenas foram alterados os valores dos parâmetros *ImgWidth* e *ImgHeight* (que passam a valer metade dos seus valores originais).

As resoluções da transformada de *Hough* e os resultados dos testes realizados utilizando metade da resolução de imagem são apresentados na tabela 6.2 e na figura 6.3. Durante estas experiências, todas as imagens de teste foram correctamente identificadas.

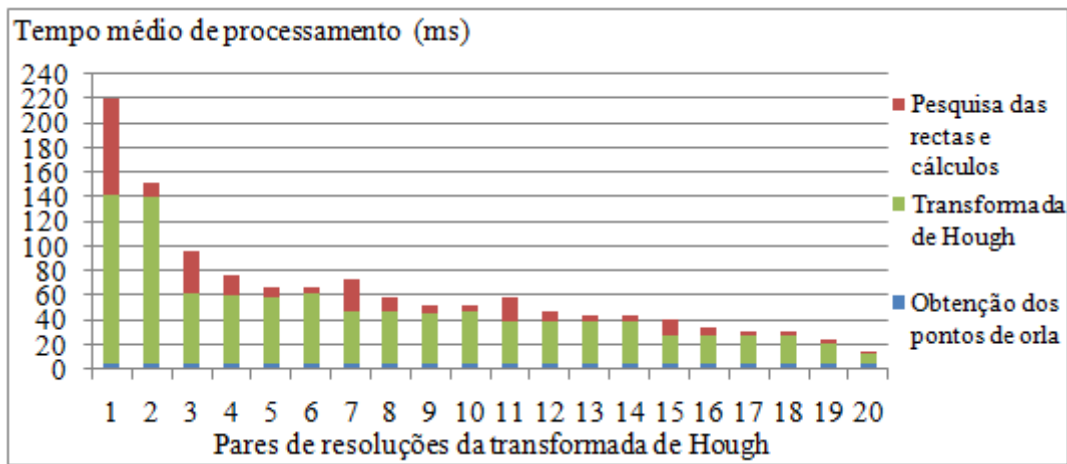


Figura 6.3: Tempo médio de processamento do algoritmo para diferentes configurações da transformada de *Hough*, utilizando metade da resolução de imagem (381 pontos transformados)

Novamente conclui-se que os erros permanecem praticamente constantes para os pares de resoluções da transformada de *Hough* testadas. Em particular, os erros absolutos médios, quando o algoritmo utiliza metade da resolução da imagem, são da mesma ordem de grandeza aos obtidos com a resolução total de imagem. Assim, caso a robustez do sistema se mantenha, a utilização dos mesmos pares de resolução da transformada seleccionados anteriormente, isto é,  $(\Delta\rho = 0,8cm, \Delta\theta = 1grau)$  e  $(\Delta\rho = 1,7cm, \Delta\theta = 2graus)$ , apresenta-se muito interessante, pois permite reduzir o tempo de processamento para cerca de metade.

Tabela 6.2: Resposta do algoritmo a 24 *frames* de teste para diferentes configurações da transformada de *Hough*, utilizando metade da resolução de imagem

Par N.º	Resolução		Erro Médio			Erro Absoluto Médio			Erro Absoluto Máximo			Desvio Padrão do Erro			Tempo de Processamento Médio (ms)
	$\Delta\rho$ (cm)	$\Delta\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	X (cm)	Y (cm)	$\theta$ (°)	
1	0,4	0,125	0,05	0,07	0,28	0,10	0,12	0,56	0,29	0,35	1,60	0,11	0,12	0,68	220
2	2,5	0,125	0,05	0,07	0,21	0,09	0,11	0,60	0,29	0,32	1,90	0,11	0,12	0,76	152
3	0,4	0,3	0,04	0,08	0,28	0,10	0,12	0,58	0,29	0,36	1,60	0,12	0,12	0,69	95
4	0,8	0,3	0,03	0,07	0,27	0,09	0,11	0,53	0,24	0,34	1,80	0,11	0,12	0,65	76
5	1,7	0,3	0,03	0,07	0,25	0,09	0,12	0,54	0,24	0,31	2,20	0,11	0,12	0,71	67
6	2,5	0,3	0,04	0,06	0,27	0,09	0,11	0,62	0,30	0,39	1,90	0,11	0,13	0,75	67
7	0,4	0,4	0,05	0,08	0,24	0,10	0,12	0,54	0,27	0,35	1,80	0,12	0,13	0,67	73
8	0,8	0,4	0,04	0,07	0,30	0,10	0,12	0,58	0,27	0,31	1,70	0,11	0,12	0,71	59
9	1,7	0,4	0,04	0,07	0,35	0,09	0,11	0,60	0,26	0,29	1,80	0,11	0,12	0,70	51
10	2,5	0,4	0,04	0,06	0,30	0,08	0,11	0,59	0,24	0,26	2,00	0,10	0,12	0,77	51
11	0,4	0,5	0,04	0,08	0,27	0,10	0,12	0,55	0,30	0,34	1,50	0,12	0,12	0,66	59
12	0,8	0,5	0,04	0,08	0,33	0,09	0,11	0,58	0,25	0,32	1,60	0,11	0,11	0,66	47
13	1,7	0,5	0,05	0,08	0,29	0,09	0,13	0,56	0,26	0,34	1,70	0,10	0,13	0,69	43
14	2,5	0,5	0,04	0,07	0,36	0,10	0,11	0,61	0,24	0,26	2,00	0,12	0,11	0,75	43
15	0,4	0,75	0,05	0,07	0,30	0,10	0,12	0,58	0,31	0,30	1,70	0,12	0,12	0,71	41
16	0,8	0,75	0,04	0,07	0,43	0,10	0,12	0,58	0,24	0,29	1,70	0,11	0,12	0,63	34
17	1,7	0,75	0,06	0,08	0,20	0,11	0,12	0,48	0,30	0,36	1,60	0,11	0,12	0,62	31
18	2,5	0,75	0,04	0,07	0,31	0,09	0,11	0,68	0,27	0,24	2,10	0,11	0,11	0,80	30
19	0,8	1	0,04	0,08	0,26	0,09	0,12	0,61	0,24	0,30	1,40	0,11	0,11	0,72	24
20	1,7	2	0,04	0,06	0,30	0,11	0,14	0,75	0,26	0,31	1,80	0,12	0,15	0,83	14

### 6.1.1 Testes de robustez

A robustez do sistema de detecção a imperfeições, degradação e ocultação do marcador é um aspecto de extrema importância, principalmente quando aplicado a ambientes com características industriais.

Um dos testes realizados ao sistema consistiu na ocultação parcial do marcador, corrompendo as *frames* recebidas com manchas circulares. O facto destas manchas serem pretas corresponde, para a tonalidade do marcador em uso, ao caso mais desfavorável. Como o seu contraste com o marcador branco é elevado, a borda da mancha é quase sempre perfeitamente detectada como ponto de orla, o que pode dificultar a correcta identificação do verdadeiro marcador.

Na figura 6.4 encontra-se um exemplo da ocultação simultânea dos 3 vértices do marcador. Visto que foram detectados suficientes pontos de orla o marcador é correctamente identificado.

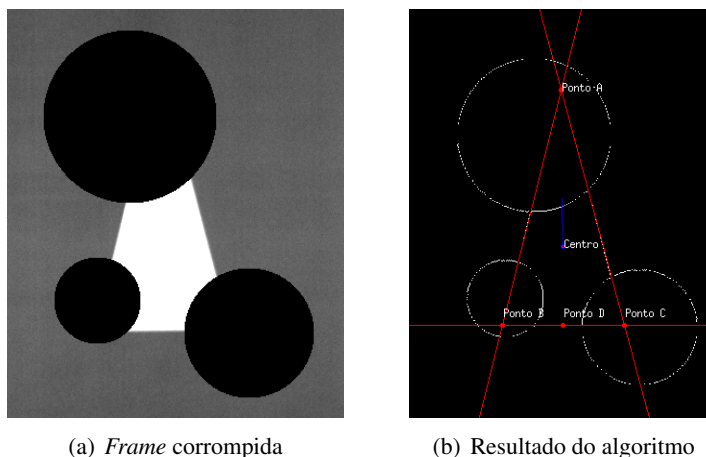


Figura 6.4: Exemplo de ocultação dos 3 vértices do marcador

No exemplo da figura 6.5 a mancha cobre a maior parte da área do marcador. Mesmo assim, o algoritmo é capaz de reconstruir e detectar correctamente a presença de um marcador na imagem.

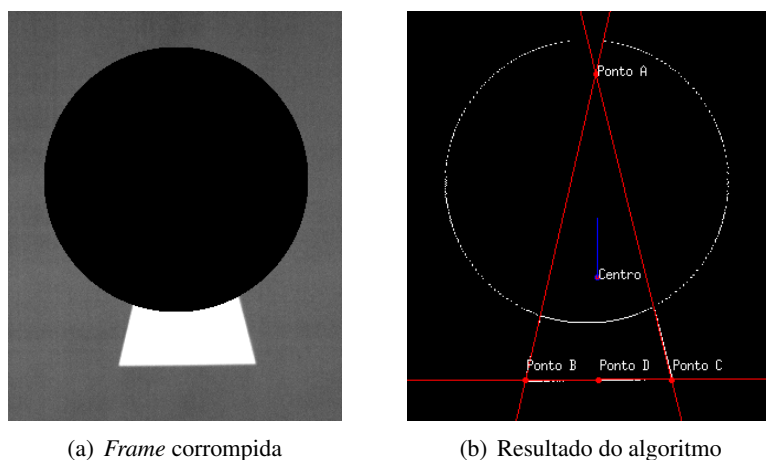


Figura 6.5: Exemplo de ocultação de grande parte da área do marcador



Apesar de se pretender um sistema flexível e capaz de superar a ocultação parcial do marcador, é fundamental que tal característica não afecte a sua integridade. Na figura 6.6 exemplifica-se um caso que poderia conduzir à falsa detecção de um marcador. A mancha introduzida esconde totalmente o lado mais curto do marcador, pelo que uma das rectas seleccionada pelo algoritmo não corresponde à sua forma geométrica original. Apesar de as rectas detectadas definirem uma espécie de triângulo, a insuficiência de semelhança com o marcador conhecido faz com que os testes de validação rejeitem correctamente esta amostra.

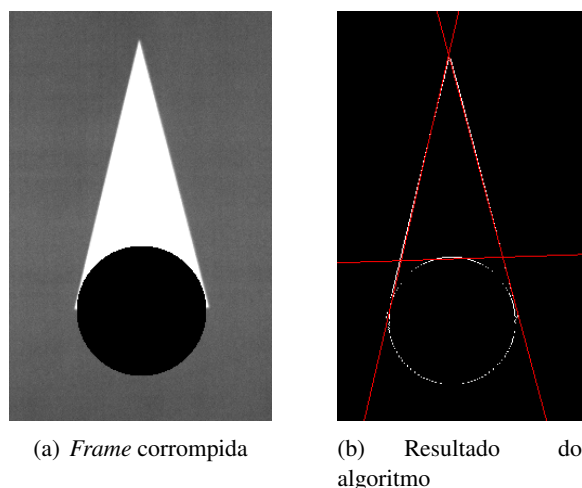


Figura 6.6: Exemplo da rejeição de falso positivo

### 6.1.2 Testes de ruído

Nestas experiências, é adicionado ruído com distribuição gaussiana, de média nula, a uma dada percentagem de pixels da *frame*, o que, apesar de não corresponder ao tipo de ruído esperado em ambiente industrial, constitui um método sistemático de degradação da imagem.

Foram realizados testes com vários níveis de ruído (vários valores de desvio padrão) e diferentes percentagens de pixels corrompidos para ambas as resoluções da transformada de *Hough* seleccionadas na secção 6.1. Para cada um dos casos, foram realizados 50 ensaios sobre uma dada *frame* de teste e verificada a resposta do algoritmo através da análise do erro da estimativa e do número de amostras não detectadas.

Alguns dos resultados obtidos utilizando a resolução total de imagem encontram-se nas figuras 6.7 a 6.10.

Facilmente se conclui que o erro das várias estimativas mantêm-se praticamente constante para os vários níveis de ruído em que o algoritmo detecta o marcador. Para níveis de ruído elevados, o algoritmo, ao invés de retornar estimativas mais distantes dos valores reais, pura e simplesmente deixa de detectar o marcador.

Comparando os valores de erro para ambas as resoluções da transformada de *Hough*, verifica-se que a utilização de uma resolução mais rápida, isto é,  $\Delta\rho = 1,7cme\Delta\theta = 2\text{grau}$ , conduziu a

resultados bastante semelhantes, sendo apenas agravado o erro da estimativa da orientação em cerca de meio grau.

Nas figuras 6.11 a 6.14 apresentam-se os resultados obtidos em testes semelhantes aos anteriores, mas utilizando metade da resolução de imagem. Como o ruído é aplicado antes da re-amostragem da *frame*, ao se realizar a média dos 4 pixels vizinhos o efeito do ruído gaussiano é atenuado. Este facto justifica que esta resolução de imagem pareça mais robusta a este tipo de degradação (o algoritmo consegue detectar o marcador na presença de níveis superiores de ruído), o que não é necessariamente verdade. Apesar disso, verifica-se novamente que o erro das estimativas é mantido em valores próximos aos obtidos para imagens não corrompidas, sendo que com o aumento dos níveis de ruído verifica-se, como seria de esperar, um aumento do número de amostras não detectadas.

Em suma, verifica-se que, para as diferentes resoluções da imagem e da transformada de *Hough* testadas, podemos esperar uma boa precisão da estimativa obtida sempre que o algoritmo detecta um marcador. Os testes de ruído realizados mostram que o aumento da degradação da imagem diminui a probabilidade de detecção do marcador, reflectindo a robustez do sistema. Nos casos em que um marcador é detectado, o erro associado não sofre grande alteração.

Sendo assim, das configurações do sistema de detecção dos marcadores visuais apresentadas, optou-se por utilizar aquela que minimiza o tempo de processamento do algoritmo ( $\Delta\rho = 1,7\text{cm}$  e  $\Delta\theta = 2\text{grau}$ , usando metade da resolução de imagem).

### 6.1.3 Velocidade máxima do robô que permite detecção

A velocidade máxima a que o robô se desloca deve permitir a correcta identificação do marcador visual. Na tabela 6.3 encontram-se o número de detecções que o algoritmo realiza para diferentes velocidades do robô.

Tabela 6.3: Número de vezes que o algoritmo detecta o marcador visual para diferentes velocidades do robô

	Velocidade do robô (cm/s)						
	20	25	30	40	50	70	80
Marcador em perfeitas condições	14	12	9	6	5	5	0
Marcador degradado	14	12	9	6	1	0	0

Verifica-se que a degradação do marcador limita a velocidade máxima do robô que ainda permite a sua detecção. Por outro lado, para velocidade na ordem dos 30 cm/s, o marcador é identificado cerca de 9 vezes, permitindo a correcção da pose do veículo e a tolerância a falhas de comunicação.

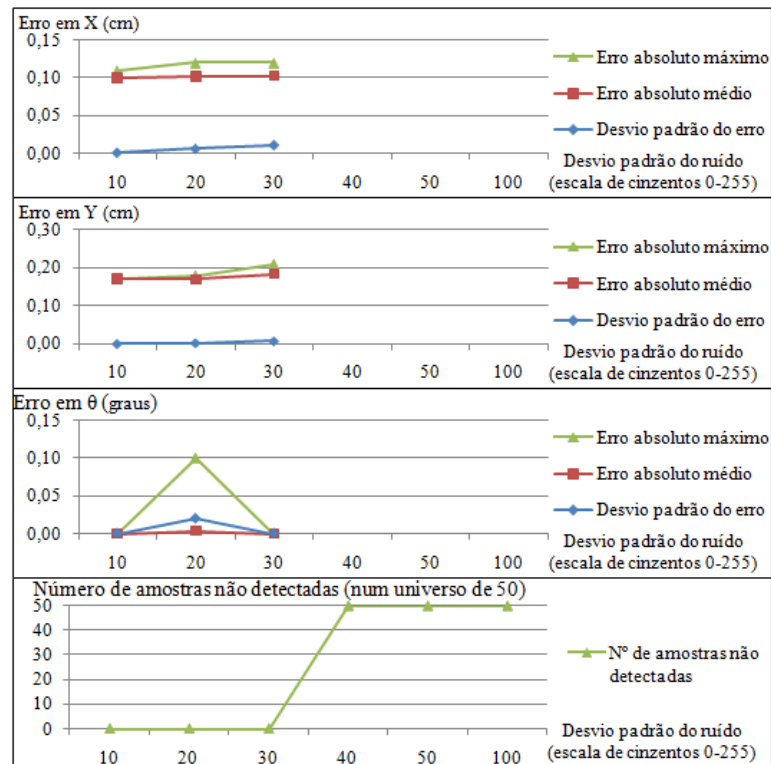


Figura 6.7: Resultados dos testes de degradação a 75% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 0,8\text{cm}, \Delta\theta = 1\text{grau}$ ), utilizando a resolução total de imagem

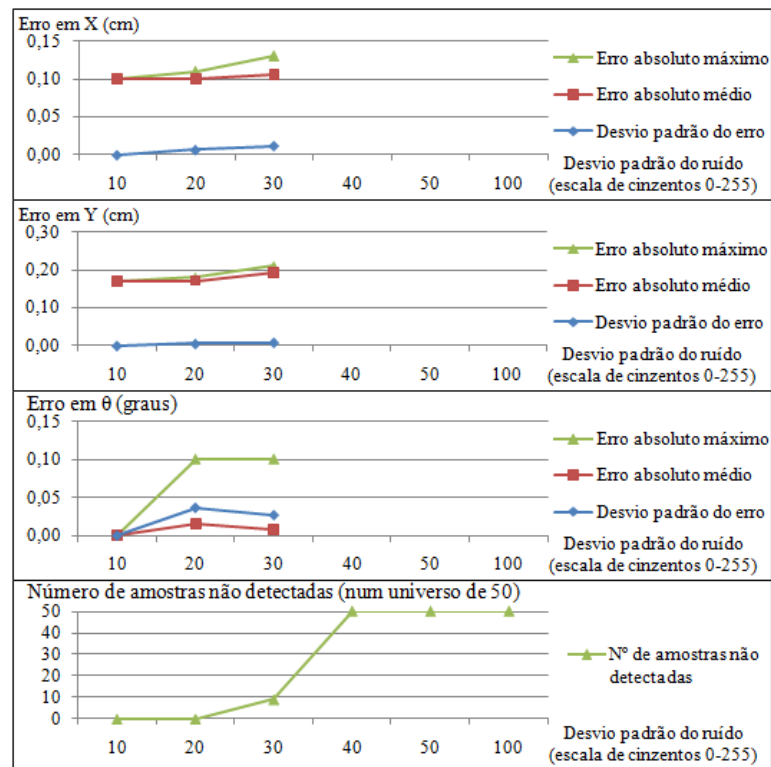


Figura 6.8: Resultados dos testes de degradação a 100% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 0,8\text{cm}, \Delta\theta = 1\text{grau}$ ), utilizando a resolução total de imagem

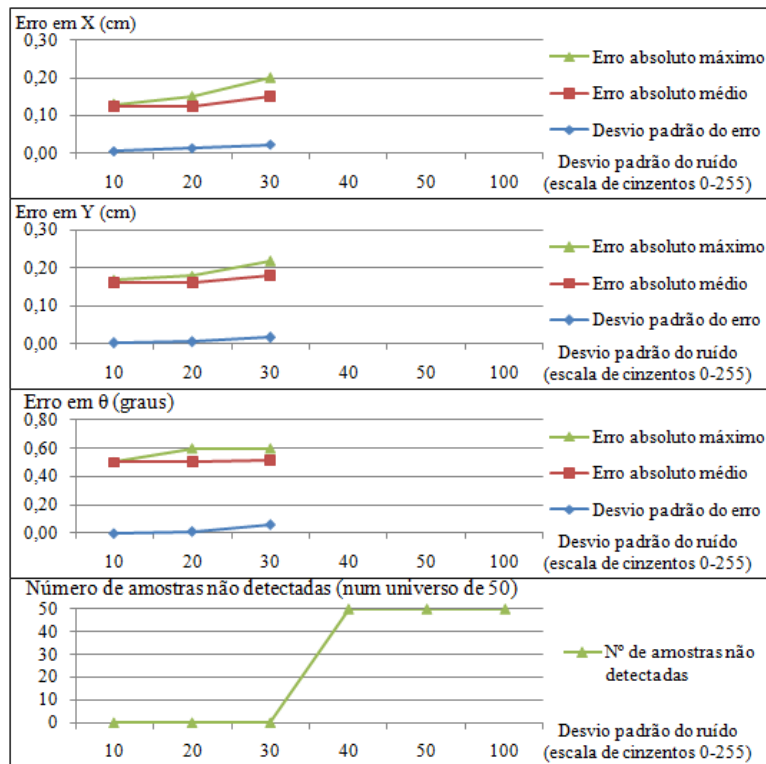


Figura 6.9: Resultados dos testes de degradação a 75% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 1,7\text{cm}$ ,  $\Delta\theta = 2\text{grau}$ ), utilizando a resolução total de imagem

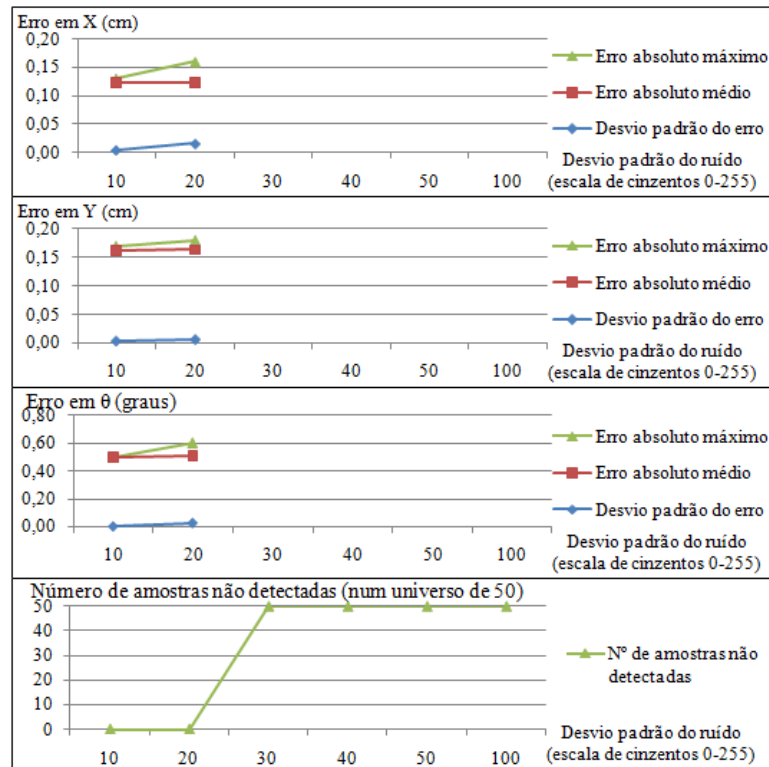


Figura 6.10: Resultados dos testes de degradação a 100% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 1,7\text{cm}$ ,  $\Delta\theta = 2\text{grau}$ ), utilizando a resolução total de imagem

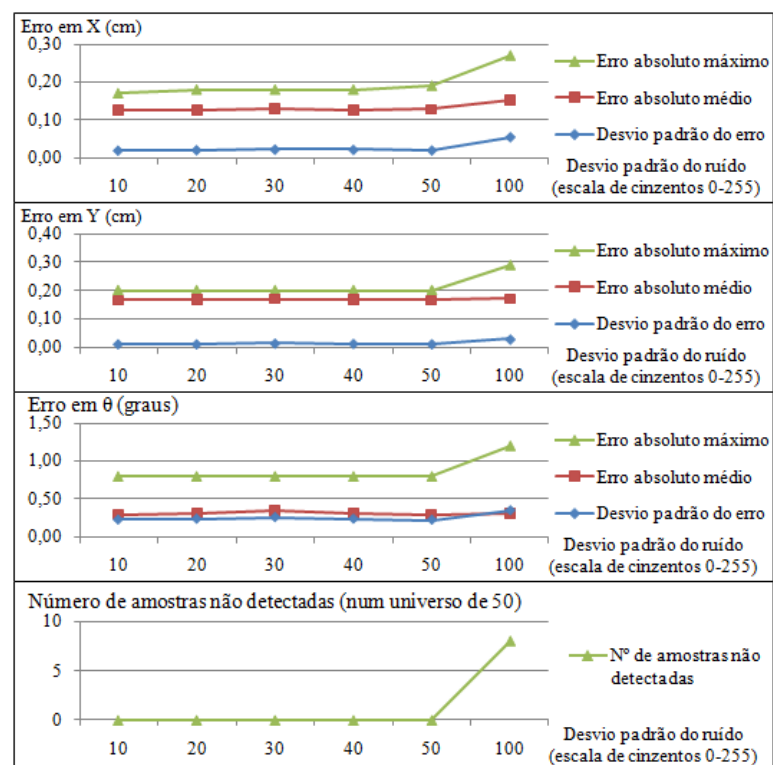


Figura 6.11: Resultados dos testes de degradação a 75% dos pixeis para a resolução da transformada de Hough ( $\Delta\rho = 0,8\text{cm}$ ,  $\Delta\theta = 1\text{grau}$ ), utilizando metade da resolução de imagem

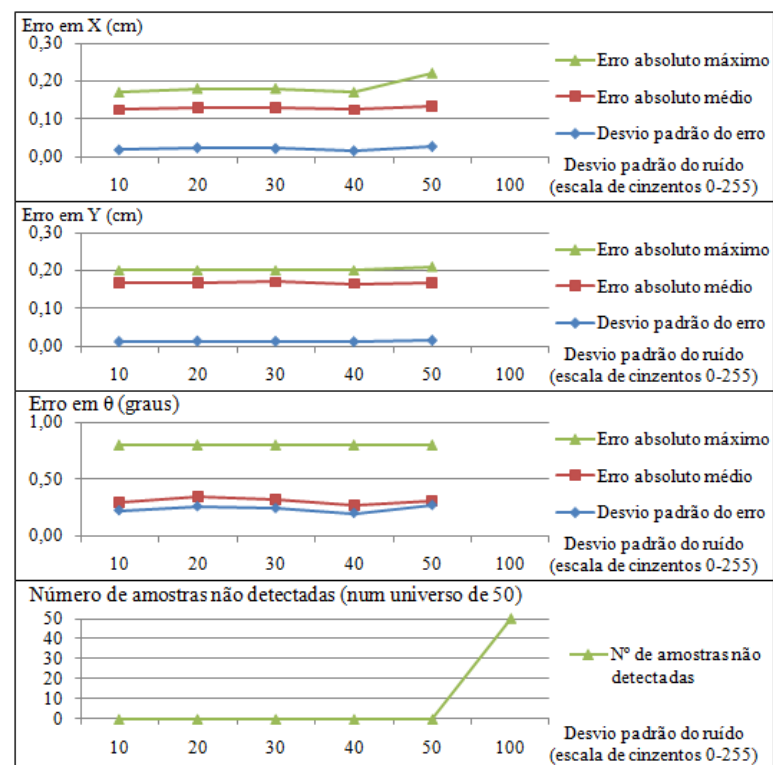


Figura 6.12: Resultados dos testes de degradação a 100% dos pixeis para a resolução da transformada de Hough ( $\Delta\rho = 0,8\text{cm}$ ,  $\Delta\theta = 1\text{grau}$ ), utilizando metade da resolução de imagem

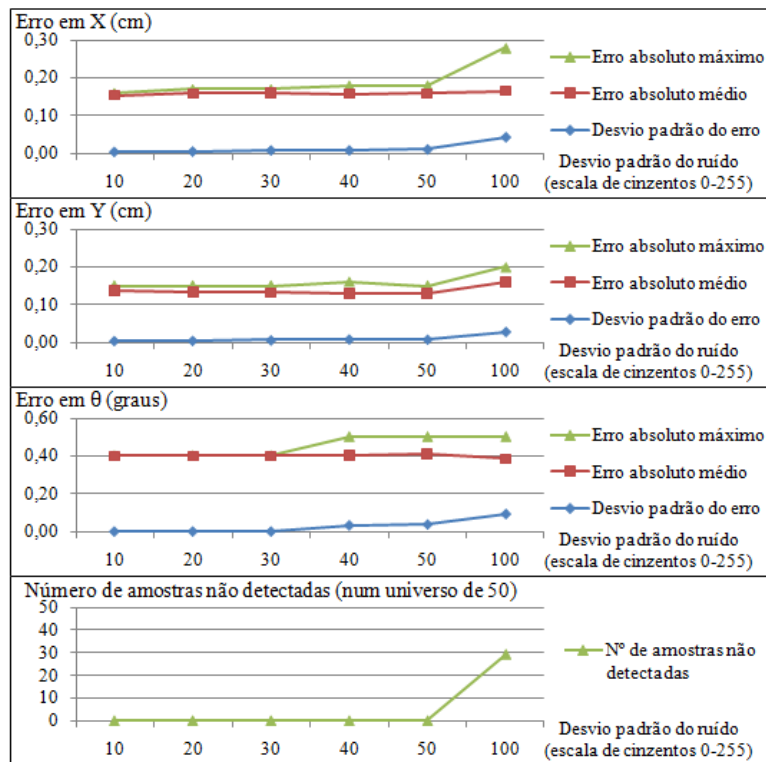


Figura 6.13: Resultados dos testes de degradação a 75% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 1,7\text{cm}$ ,  $\Delta\theta = 2\text{grau}$ ), utilizando metade da resolução de imagem

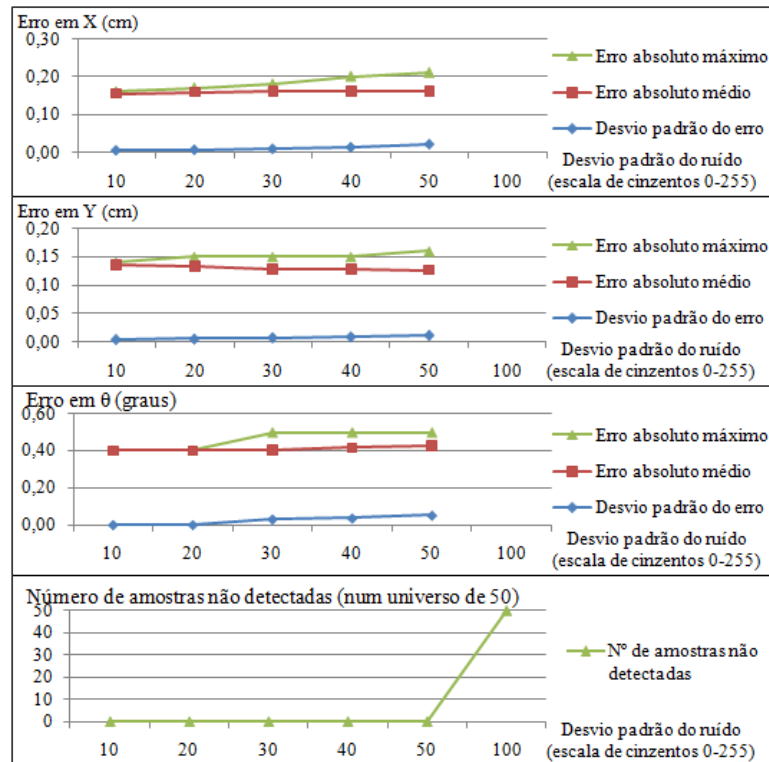


Figura 6.14: Resultados dos testes de degradação a 100% dos pixels para a resolução da transformada de Hough ( $\Delta\rho = 1,7\text{cm}$ ,  $\Delta\theta = 2\text{grau}$ ), utilizando metade da resolução de imagem

### 6.1.4 Percurso de teste

A fim de validar o sistema de localização com recurso a marcadores visuais, foram realizados testes num percurso semelhante ao retratado na figura 6.15. A execução da trajectória utiliza um procedimento de deslocamento entre pontos consecutivos, implementado na função *GoToXYTheta*. O robô partiu inicialmente de A, com orientação inversa ao do eixo  $Y$ , isto é,  $-90^\circ$ , terminando o percurso no ponto E do mapa com a mesma orientação.

Foram colocados marcadores visuais perto do ponto A, com o objectivo de inicializar o robô com uma estimativa correcta da pose, e do ponto E, para avaliar o erro no fim da trajectória. Inicialmente, o percurso foi repetido 5 vezes utilizando apenas a odometria para localizar o robô. De seguida, foram repetidas as experiências, mas já com o veículo a localizar-se utilizando o Filtro de *Kalman* Estendido (EKF). Durante este teste, o único marcador visual utilizado pelo robô para se localizar foi  $M_1$ , colocado em  $(x=2,52\text{ m}; y=2,72\text{ m}; \theta=0^\circ)$ . Os restantes marcadores serviram apenas para comparar a pose estimada pelo robô com a real.

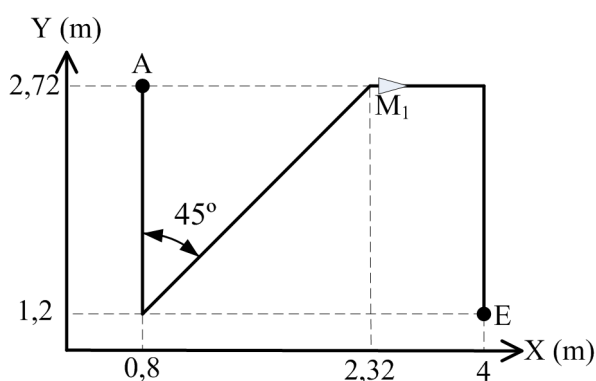


Figura 6.15: Percurso de teste executado e localização do marcador visual

Nas figuras 6.16 e 6.17 encontram-se, respectivamente, a estimativa da trajectória executada pelo robô dada pela odometria e a evolução temporal de cada componente do estado do robô.

Os valores dados pelos marcadores visuais que se encontram sobre o eixo do  $X$  correspondem aos instantes de tempo em que nenhum marcador é visível e por isso os seus valores são definidos como nulos.

Verifica-se que a trajectória executada pelo veículo, quando este se localiza apenas pela odometria, difere daquela que está disponível quando o robô se desloca sobre um dos 3 marcadores visuais. A não correcção da pose do veículo por um método de localização absoluta acabaria por fazer a estimativa do estado do robô divergir do seu valor real.

De salientar que durante esta trajectória essa diferença não é muito visível, fruto de uma boa calibração da odometria. Para além disso, a pose dada pela odometria parece, no final da trajectória, encontrar-se mais próxima dos valores estimados pelos marcadores do que na proximidade do marcador  $M_1$ , o que sugere que as características do percurso proporcionam o cancelamento de alguns dos erros da odometria.

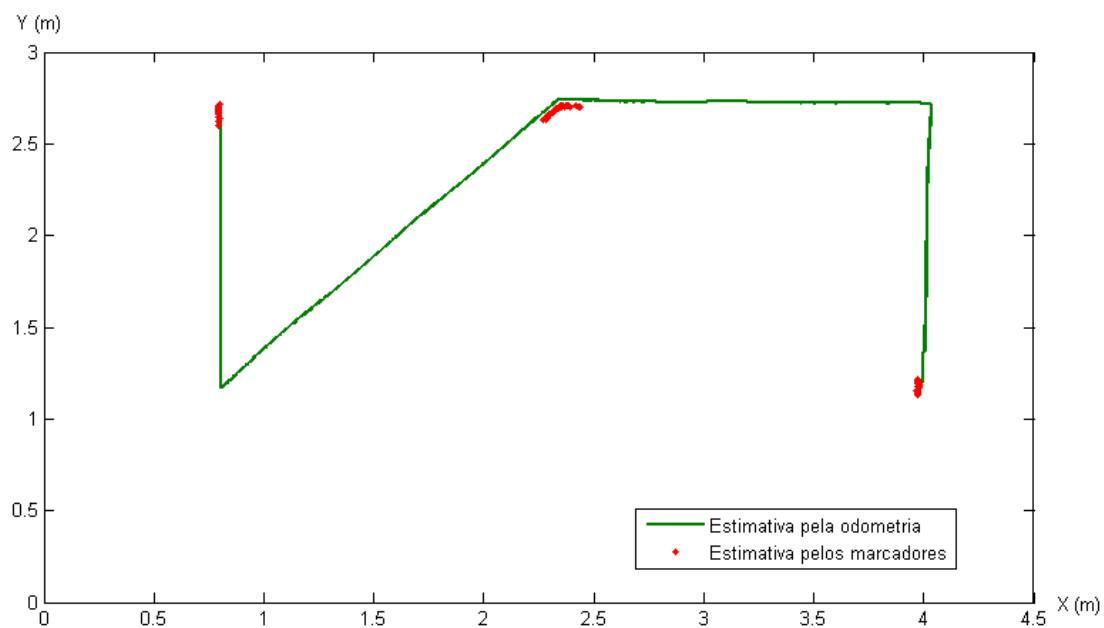


Figura 6.16: Estimativa da trajetória executada pelo robô, utilizando apenas a odometria

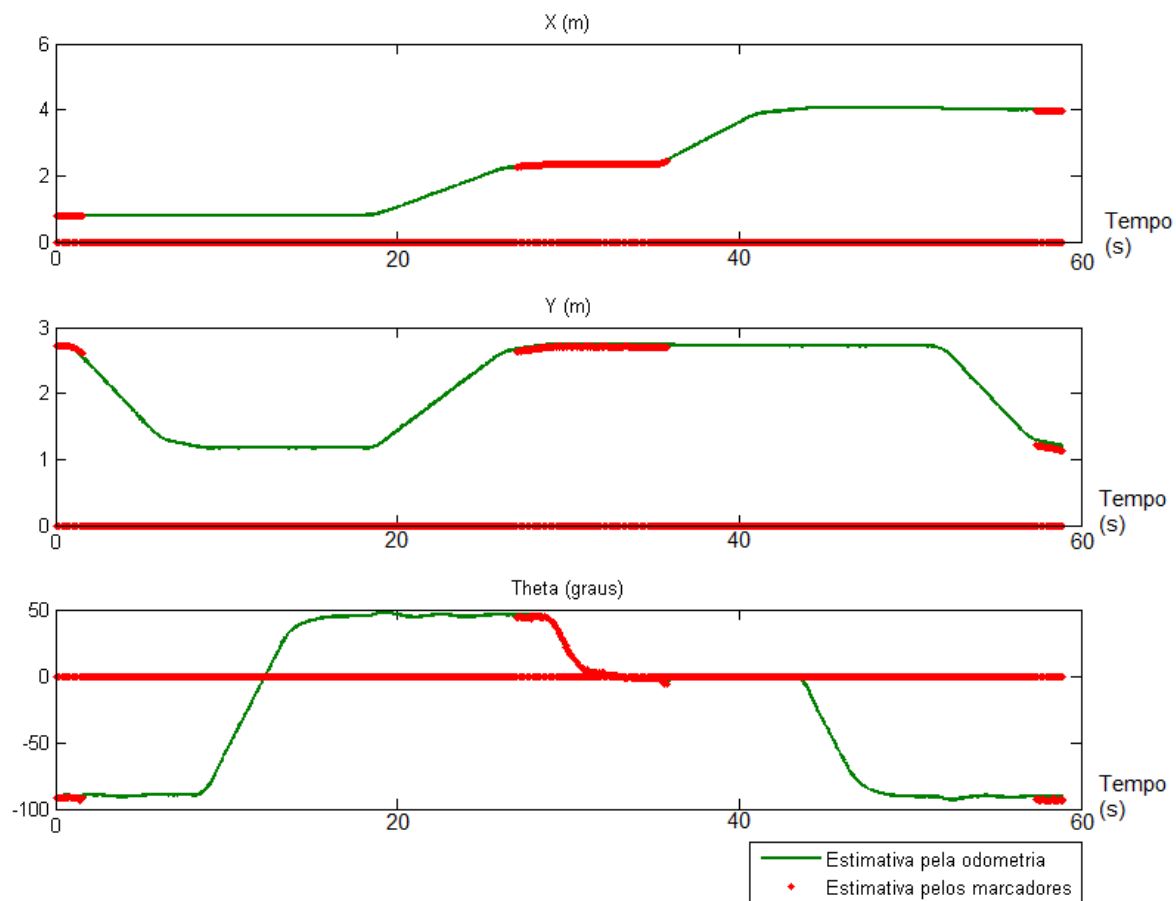


Figura 6.17: Evolução temporal da estimativa da pose do robô, utilizando apenas odometria



Nas figuras 6.18 e 6.19 exemplifica-se a estimativa da trajectória executada pelo robô utilizando o Filtro de *Kalman* Estendido para fundir as medidas fornecidas pelo marcador M1 com a odometria, bem como a evolução temporal de cada componente do estado do robô.

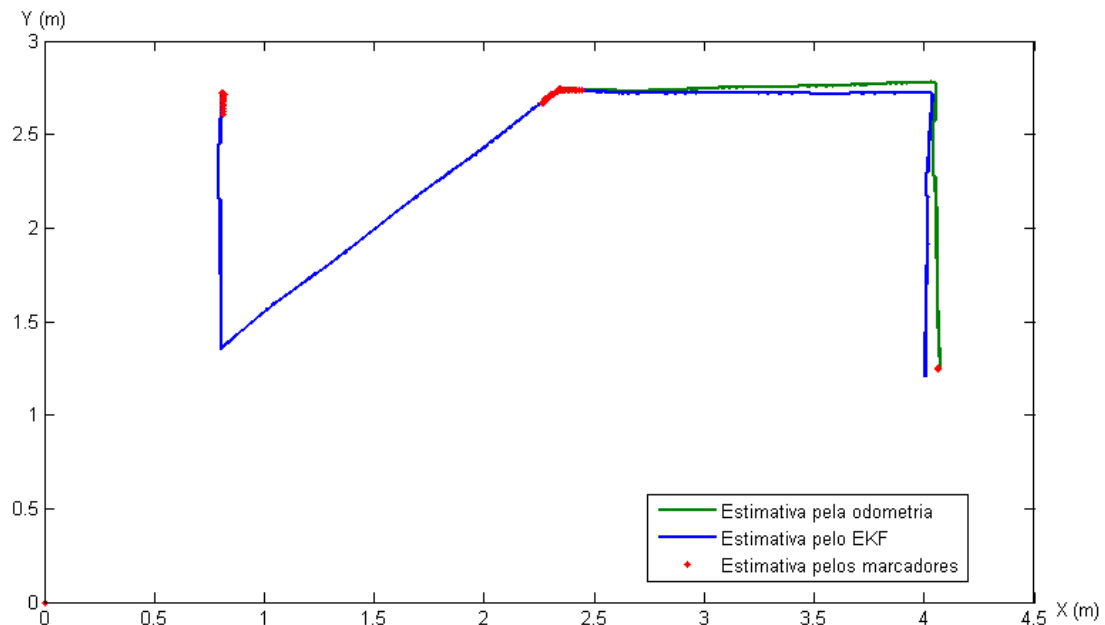


Figura 6.18: Estimativa da trajectória executada pelo robô, utilizando o EKF

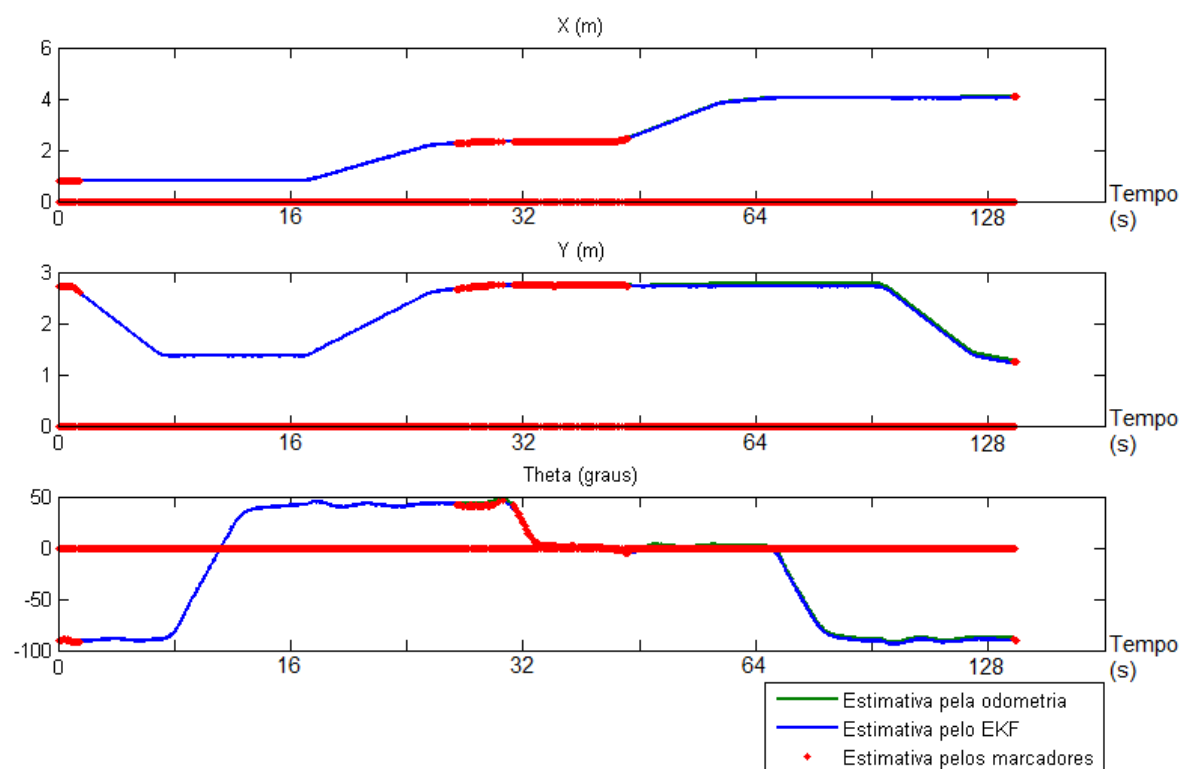


Figura 6.19: Evolução temporal da estimativa da pose do robô, utilizando o EKF

Novamente, os valores dados pelos marcadores visuais que se encontram sobre o eixo do X correspondem aos instantes de tempo em que nenhum marcador é visível e, por isso, não são usados na fusão sensorial.

Uma primeira observação dos gráficos obtidos com a fusão sensorial parece sugerir que a pose fornecida apenas pela odometria para o mesmo movimento do robô é de melhor qualidade. No entanto, ao se analisar a figura 6.16 já se verificou que os erros da odometria devem se cancelar, pois o erro no final da trajectória é inferior ao verificado por volta do meio da mesma, na proximidade do marcador M1. Analisando a figura 6.19 conclui-se que a estimativa dada pelo Filtro de *Kalman* Estendido segue os valores fornecidos pelos marcadores visuais, corrigindo adequadamente o estado do robô sempre que uma medida está disponível.

De modo a comparar a trajectória executada utilizando somente o método de localização relativa (odometria) com a efectuada recorrendo à fusão de informação do método de localização absoluta, baseado em marcadores visuais, analisou-se os valores absolutos máximos de erro ao longo de todo o percurso de teste. A média desses valores para cada estratégia de localização encontra-se na tabela 6.4.

Tabela 6.4: Comparação do erro absoluto máximo ao longo do percurso, utilizando a odometria e a fusão de informação através do EKF (odometria e marcadores visuais)

	Erro absoluto máximo		
	X (m)	Y (m)	$\theta$ (°)
Localização baseada na odometria	0,058	0,104	4,31
Localização através do EKF	0,073	0,048	2,48

Os resultados obtidos são bastante animadores e comprovam a melhoria alcançada com a utilização da fusão sensorial. Apesar do erro máximo em  $x$  ter aumentado ligeiramente, fruto provavelmente das características do percurso que possibilitam cancelamento dos erros da odometria, os erros em  $y$  e  $\theta$  melhoraram bastante com o uso do marcador visual. De salientar que a estimativa da orientação é um aspecto de grande importância em sistemas de localização baseados em medidas relativas, como a odometria. Neste caso, a inclusão de apenas um marcador permitiu reduzir o erro absoluto máximo de orientação do robô ao longo de todo o percurso de forma substancial.

Em suma, o método de localização baseado em marcadores visuais é capaz de fornecer uma estimativa precisa da pose do robô, permitindo, durante a navegação do veículo, a melhoria da estimativa da sua localização no ambiente.

## 6.2 Análise ao erro de medida do *Laser Range Finder*

O objectivo deste teste foi analisar a evolução do erro de medida do *Laser Range Finder* com a distância. Para isso, colocou-se um objecto plano perpendicularmente à direcção do feixe do *laser*. Foram realizadas experiências com o objecto posicionado a diferentes distâncias do *laser*. Para

cada uma das experiências foram guardados 128 valores de medida de modo a avaliar a dispersão dos dados e o erro médio. Os resultados obtidos encontram-se na figura 6.20.

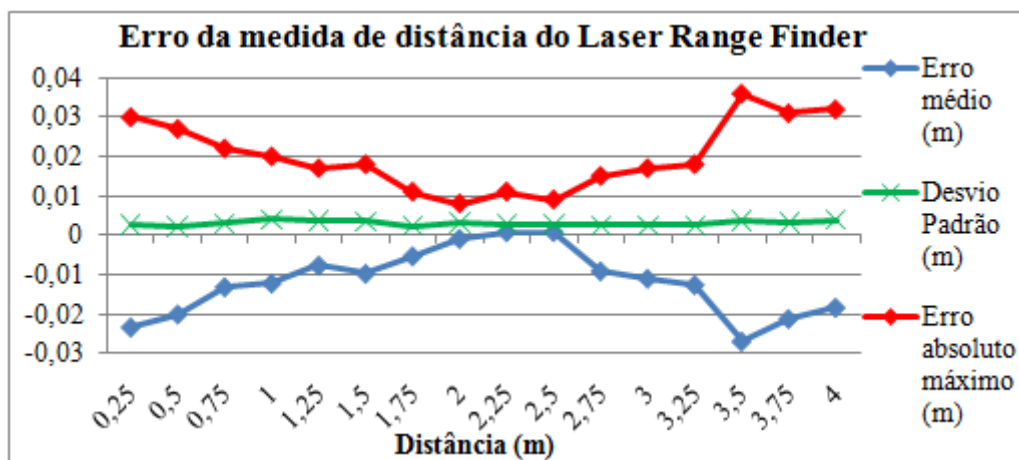


Figura 6.20: Análise ao erro de medida do *Laser Range Finder*

Deve-se assumir alguma imprecisão na medida real de distância. Mesmo assim, é possível verificar que a dispersão das medidas, dada pelo seu desvio padrão, é praticamente constante. O erro absoluto máximo, apesar de variar, manteve-se dentro dos limites indicados no *datasheet* do fabricante (ver secção 3.2.5). Já o erro médio parece indicar a existência de um desvio da medida.

### 6.3 Testes ao sistema de detecção dos marcadores em código de barras

Em relação ao sistema de detecção dos marcadores de códigos de barras é de interesse testar a sua resposta para diferentes localizações do marcador face ao *laser*. Assim, executou-se o algoritmo continuamente até se perfazerem 128 detecções bem sucedidas, isto é, identificando correctamente o código do marcador. Parte dos resultados obtidos encontram-se na tabela 6.5.

Tabela 6.5: Resultados dos testes à detecção do marcador de código de barras

Distância do marcador	Orientação do marcador	Erro Absoluto Médio			Variância do erro			Leituras com detecção de código	
		X	Y	$\theta$	X	Y	$\theta$	correcta	errada
m	°	m	m	°	$10^{-6}m^2$	$10^{-6}m^2$	$^{\circ}2$	%	%
0,15	90	0,0021	0,0068	0,643	3,79	1,35	0,606	28,6	0,67
0,2	90	0,0049	0,0086	0,803	4,37	1,43	0,936	27,2	1,91
0,4	90	0,0103	0,0142	2,061	3,12	0,60	0,568	94,1	0,00
0,6	90	0,0019	0,0205	1,239	4,38	0,32	0,780	91,4	0,71
0,7	90	0,0033	0,0259	1,985	3,48	0,23	0,872	60,7	0,00
0,2	60	0,0026	0,0062	1,195	3,67	2,17	0,753	36,4	0,28
0,6	60	0,0022	0,0182	2,818	3,73	0,15	0,630	12,3	0,10
0,7	60	0,0019	0,0217	3,132	5,88	1,07	0,809	45,2	0,00

A partir da tabela 6.5 verifica-se que o erro absoluto médio de estimativa da pose do marcador tende a aumentar com a distância, principalmente o erro segundo  $y$  e  $\theta$ . Se considerarmos um marcador com uma dada largura colocado perpendicularmente ao *laser*, quanto mais este se afasta do *laser*, menos serão os pontos medidos que lhe correspondem, influenciando a qualidade da medida. O erro segundo  $x$ , direcção da medida do laser e perpendicular ao marcador quando este está orientado com  $90^\circ$ , manteve-se maioritariamente inferior a 1 cm. Por outro lado, os valores de variância do erro permaneceram praticamente constantes.

Verificou-se que, para as dimensões do protótipo utilizado, a detecção só é possível quando o marcador se encontra aproximadamente entre 20 a 70 cm do *laser*. De salientar ainda que só para distâncias intermédias é que o marcador é detectado em quase todos os varrimentos (percentagem de identificações correctas superior a 90 %). A quantidade de falsas identificações foi reduzida (inferior a 2% das leituras do *laser*). É bom lembrar que os códigos falsamente calculados não correspondiam ao grupo seleccionado na secção 4.3, pelo que seriam despistados pelo algoritmo de localização.

Como seria de esperar, o marcador deixou de ser detectado à medida que a sua orientação era desviada da perpendicular com a direcção do feixe do *laser*. No entanto, verificam-se resultados ainda satisfatórios quando o marcador se encontra a  $30^\circ$  dessa perpendicular.

### 6.3.1 Teste em condições de funcionamento

A fim de validar as potencialidades do sistema de detecção de marcadores em relevo e comprovar a sua capacidade de fornecer uma estimativa precisa da pose do robô foram realizados testes em condições semelhantes às de funcionamento. Como o objectivo deste teste passa por provar que a pose do robô é adequadamente corrigida quando o veículo circula nas proximidades de um marcador em relevo, optou-se por inicializar o veículo com uma estimativa errada da sua pose e fazê-lo navegar próximo de um marcador, observando depois o seu comportamento.

Alguns dados sobre o teste realizado encontram-se na figura 6.21. O ponto C ( $x=1,8$  m;  $y=7$  m;  $\theta=90^\circ$ ) trata-se do local de destino. O robô parte inicialmente do ponto A ( $x=2,5$  m;  $y=0,9$  m;  $\theta=115^\circ$ ), mas a sua pose de partida foi inicializada como correspondente ao ponto B ( $x=1,8$  m;  $y=3$  m;  $\theta=90^\circ$ ). Devido a esta diferença, quando o robô detectar o marcador em relevo, o erro da estimativa da sua pose será elevado, possibilitando a clara visualização da correcção efectuada.

As figuras 6.22 e 6.23 mostram a trajectória executada e a evolução temporal do estado do robô durante a navegação com recurso ao Filtro de *Kalman* Estendido (fusão da odometria com o marcador em relevo) e a estimativa dada unicamente pela odometria durante o mesmo movimento.

Conforme se constata nos gráficos apresentados, assim que o veículo detecta o marcador em relevo o estado do robô é actualizado, convergindo para a pose fornecida pelo marcador. Esta correcção permitiu que o robô alcançasse de forma precisa o seu destino, apesar de ter sido inicializado propositadamente com uma estimativa errónea da sua pose. Durante a execução desta trajectória registou-se a estimativa da pose do robô dada unicamente pela odometria. Como seria de esperar, a não correcção desta estimativa com os dados fornecidos pela detecção do marcador em relevo, leva a que a odometria avalie incorrectamente a localização do veículo.

Com este teste verifica-se que o sistema baseado em marcadores em relevo é capaz de fornecer uma medida precisa da pose do robô durante a sua navegação.

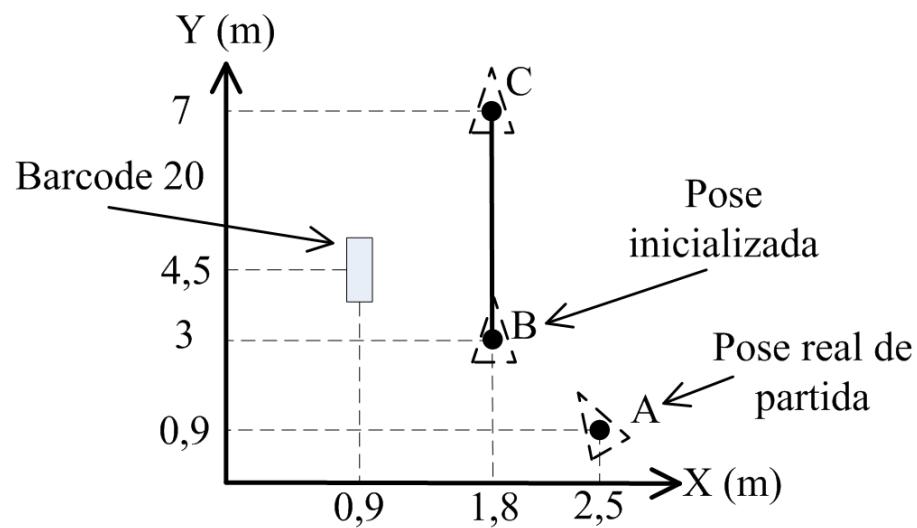


Figura 6.21: Percurso de teste dos marcadores em relevo

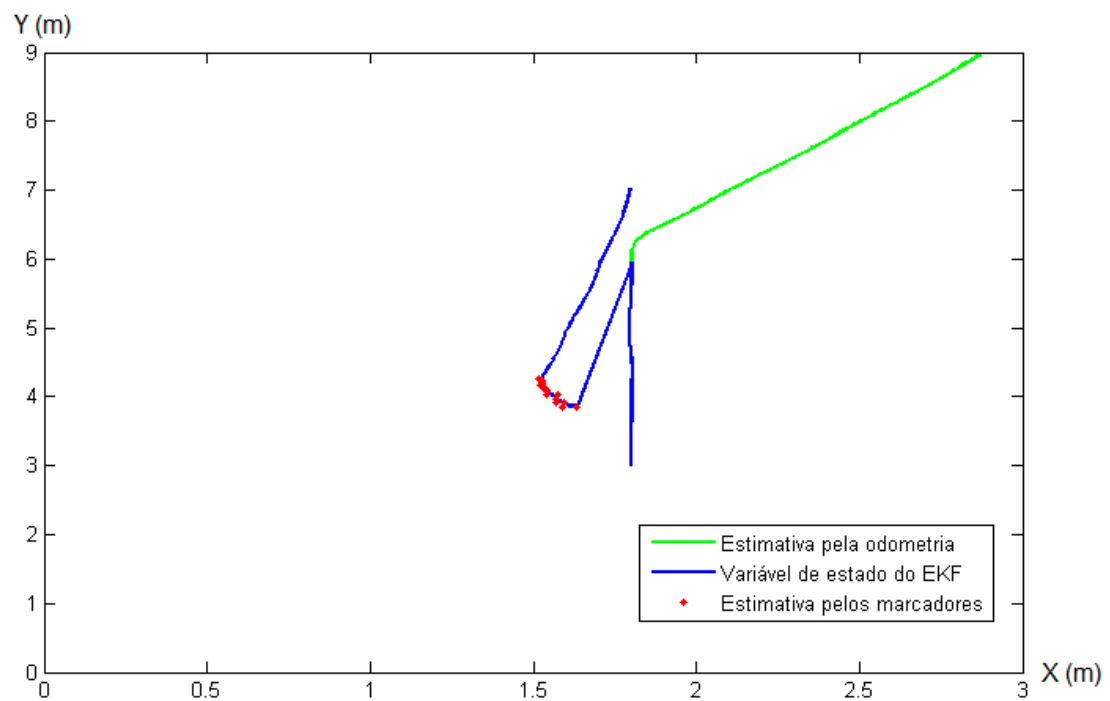


Figura 6.22: Trajectória estimada e executada pelo robô - Percurso de teste dos marcadores em relevo

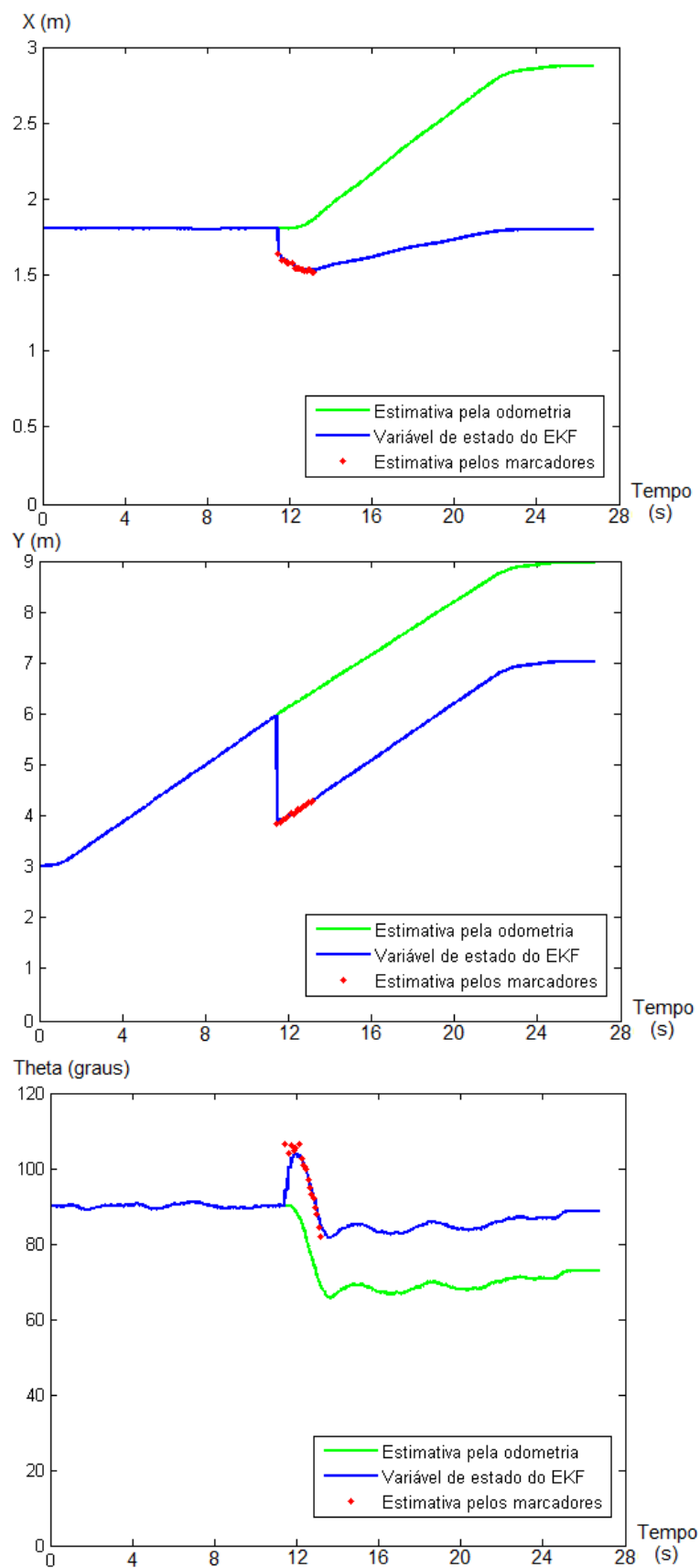


Figura 6.23: Evolução temporal das posições reais e estimadas pelo robô - Percurso de teste dos marcadores em relevo

## 6.4 Teste à detecção de obstáculos

A segurança é um factor de extrema importância no desenvolvimento de sistemas autónomos. Assim, de modo a validar a funcionalidade de detecção de obstáculos do robô móvel desenvolvido, ordenou-se ao veículo a execução de uma trajectória rectilínea segundo X nas condições apresentadas na figura 6.24. O robô parte da origem orientado segundo o eixo do X com destino ao ponto  $(x, y) = (5, 0)$ . No entanto, a presença de um obstáculo nesse ponto vai obrigá-lo a abrandar e parar antes do alvo.

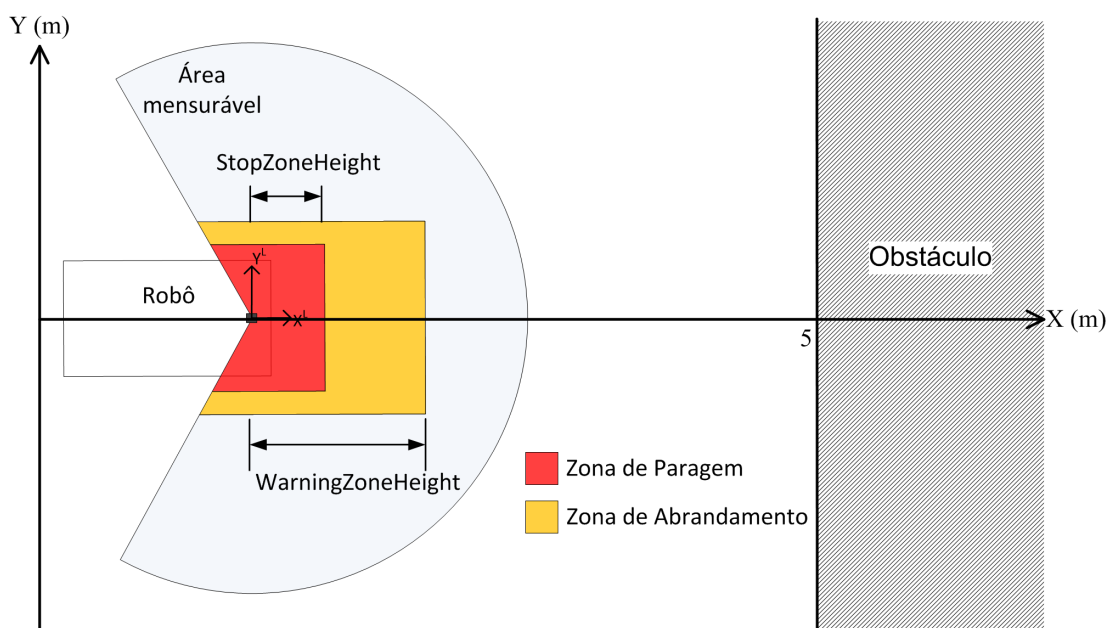


Figura 6.24: Teste do sistema de detecção de obstáculos

As figuras 6.25 e 6.26 apresentam a evolução temporal da posição na coordenada X e da velocidade linear do robô ao longo da trajectória de teste do sistema de detecção de obstáculos. Neste caso, o valor de *StopZoneHeight* e de *WarningZoneHeight* utilizados foram, respectivamente, de 0,6 e 1,5 metros. Como o referencial do *Laser* e do robô distam, segundo o eixo X de 0,2 metros, o robô inicia a redução de velocidade após percorrer aproximadamente 3,30 metros, isto é, a cerca de 1,7 metros do obstáculo. Inicialmente a velocidade do veículo é de 25 cm/s, mas, na zona de abrandamento, passa a 10 cm/s. O veículo pára quando se encontra em  $X=4,20$  m, ou seja, a 80 cm do obstáculo.

Como se verifica, o robô é plenamente capaz de detectar obstáculos dentro da gama de medição do *Laser Range Finder* e assim alterar o seu comportamento de modo a garantir uma navegação em segurança. Assim que as zonas de segurança definidas se encontram livres, o robô retorna ao seu estado de operação normal, podendo deslocar-se à velocidade nominal desejada.

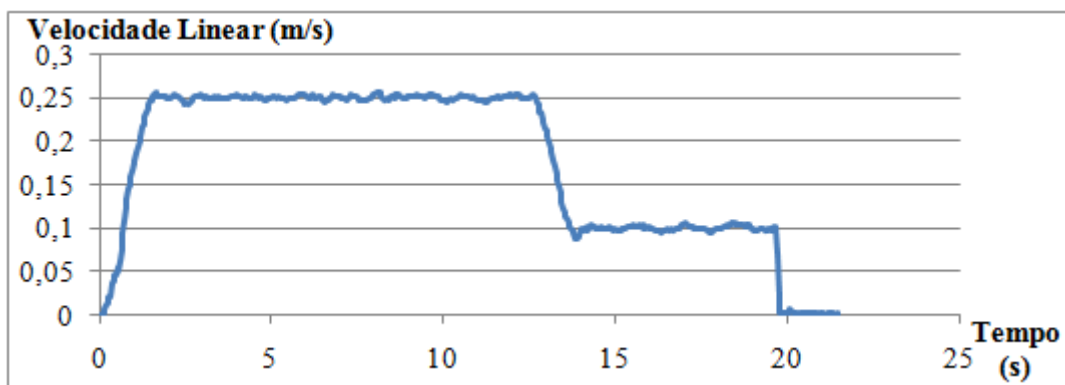


Figura 6.25: Evolução temporal da velocidade linear do robô durante o teste do sistema de detecção de obstáculos

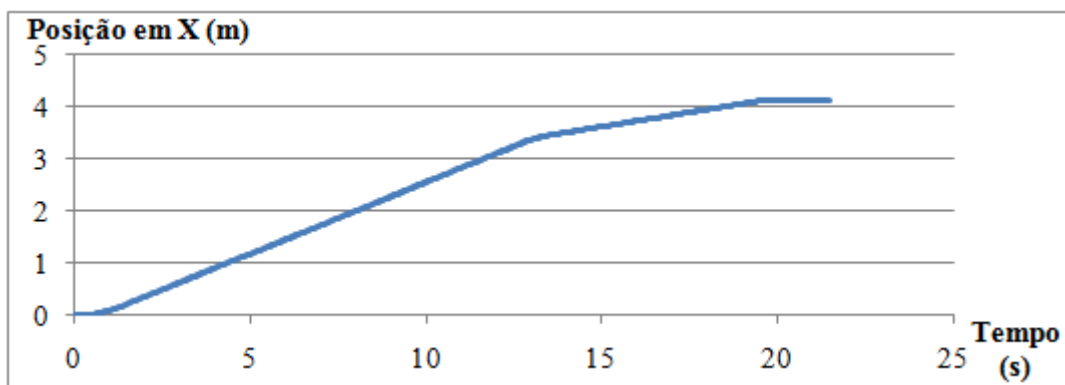


Figura 6.26: Evolução temporal da posição X do robô durante o teste do sistema de detecção de obstáculos



## Capítulo 7

# Conclusões

Face ao resultado final desta dissertação, podemos concluir que os objectivos inicialmente definidos foram alcançados. Na verdade, foram propostos dois sistemas distintos de localização absoluta de robôs móveis baseados em marcadores artificiais.

Foi desenvolvido um sistema de identificação de marcadores, com forma geométrica triangular, através de visão artificial. Um método de codificação unívoca dos marcadores foi apresentado e implementado. Os testes realizados comprovaram a robustez do sistema à degradação do marcador, despistando falsas identificações, bem como a precisão do cálculo da sua pose. Houve uma atenção especial em reduzir o poder computacional exigido pela aplicação que executa os algoritmos de detecção do marcador.

Complementarmente, foi apresentado um método de localização baseado no reconhecimento de marcadores em relevo semelhantes a um código de barras. Apesar de não terem sido realizados ainda percursos de teste, ficou provada a sua capacidade de prover uma estimativa da localização do robô móvel.

As soluções de localização baseadas em marcadores revelam-se muito vantajosas para aplicações com necessidades de flexibilidade e reconfiguração frequente. A utilização de um Filtro de *Kalman* Estendido permitiu fundir informação de diversas medidas e melhorar a estimativa da pose do robô. Com este método poderão ser acrescentadas novas soluções de localização que venham a ser estudadas.

Para evitar colisões contra pessoas e bens, foi implementada uma tecnologia de detecção de obstáculos baseada num sensor *laser*.

Durante este projecto instalou-se a estrutura base de um robô móvel industrial, o que permitiu testar os sistemas desenvolvidos num ambiente com características mais próximas da realidade. O custo moderado das soluções propostas abrem perspectivas de aplicação comercial.

## 7.1 Trabalho Futuro

Com vista à continuidade deste projecto são apresentadas de seguida algumas sugestões de trabalho futuro.

Em relação ao sistema de detecção de marcadores visuais sugere-se o estudo de um algoritmo de identificação dos segmentos alternativo à transformada de *Hough*. O objectivo passa por avaliar a precisão alcançada com um método computacionalmente mais leve.

Quanto ao sistema de localização baseado em marcadores em relevo, propõe-se o estudo das dimensões dos segmentos do marcador que aumentem a precisão da localização e a distância máxima a que esta é possível. As aplicações devem ser melhoradas no sentido de aumentar o número de códigos utilizáveis. Testes complementares de robustez e precisão deverão também ser realizados, incluindo testes em diferentes condições de funcionamento.

Propõe-se igualmente como trabalho futuro, a calibração da odometria recorrendo a um método mais completo, como por exemplo a técnica *UMBMark*. Um progresso interessante seria o desenvolvimento de rotinas de calibração automática da odometria quando o robô se encontra na presença de marcadores.

O sistema de detecção de obstáculos pode ser melhorado a fim de, para além de detectar obstáculos num maior número de situações, permitir o contorno dos mesmos. Para isso sugere-se a implementação de um método que mapeie a área envolvente ao robô numa grelha cartesiana bidimensional. A criação de um histograma sobre essa estrutura permitiria evitar certas colisões durante a rotação do veículo. A aplicação de um método como o *Vector Field Histogram* (VFH) possibilitaria controlar o robô de modo a alcançar o seu destino sem colidir.

Por fim, sugere-se a melhoria da interface com o utilizador da aplicação de controlo do veículo e o desenvolvimento de funcionalidades extra ao nível da navegação e planeamento de trajectórias.

## Anexo A

# Intersecção de rectas definidas por dois pontos

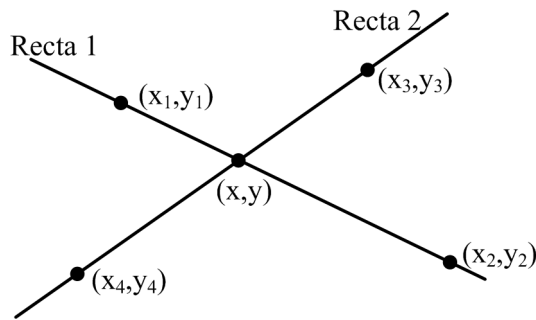


Figura A.1: Ponto de intersecção de duas rectas definidas por dois pontos

O ponto  $(x, y)$  de intersecção de duas rectas definidas por dois pontos (ver figura A.1) pode ser obtido através do seguinte cálculo baseado em determinantes [55]:

$$x = \frac{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} x_3 - x_4}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}}, \quad y = \frac{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} y_3 - y_4}{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix}} \quad (\text{A.1})$$



# Referências

- [1] AGV's - Segurança e Navegação. Disponível em <http://www.ek-automation.com/products/automated-guided-vehicles/navigation.html>. Último acesso a 9 de Março de 2010.
- [2] E'nsor - Egemin Navigation System on Robot AGV. Disponível em <http://www.egeminusa.com/pages/software/ensor.html#>. Último acesso a 9 de Março de 2010.
- [3] Hugo Alves. Navegação e Controlo de um Veículo Móvel. Tese de Mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2010.
- [4] KIVA Systems. Disponível em <http://www.kivasystems.com/>. Último acesso a 18 de Março de 2010.
- [5] Visual Robots - Peek into the Future of Material Handling. Disponível em <http://www.automation.com/resources-tools/articles-white-papers/articles-by-bill-lydon/seegrid-visual-robots>. Último acesso a 18 de Março de 2010.
- [6] AGVs Make The Workplace safer. Disponível em <http://www.hksystems.com/agv/safety.cfm>. Último acesso a 18 de Março de 2010.
- [7] Seegrid Corporation - Industrial Mobile Robots. Disponível em <http://www.seegrid.com>. Último acesso a 18 de Março de 2010.
- [8] How Color Cameras Work, Abril 2010. Disponível em [http://www.theimagingsource.com/en\\_US/publications/whitepapers/](http://www.theimagingsource.com/en_US/publications/whitepapers/).
- [9] Armando Sousa. *Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão - Aplicações no Domínio do Futebol Robótico*. Tese de Doutoramento em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2003.
- [10] G. G. Savii. Camera Calibration Using Compound Genetic-Simplex Algorithm. *Journal of Optoelectronics and Advanced Materials*, 6(4):1255–1261.
- [11] R. Siegwart e I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 2004.
- [12] Rui Paulo Rocha. Estado da Arte da Robótica Móvel em Portugal, Março 2001. Disponível em <http://mail.isr.uc.pt/~mrl/admin/upload/37.pdf>.

- [13] J. Borenstein, H. R. Everett, e L. Feng. *"Where am I? - Systems and Methods for Mobile Robot Positioning"*. The University of Michigan, 1996.
- [14] The Basics of Automatic Guided Vehicle Systems. Disponível em <http://www.agvsystems.com/basics/index.htm>. Último acesso a 26 de Fevereiro de 2010.
- [15] Pedro M. Carvalho, A. Paulo Moreira, e Paulo José Costa. Small and Low-Cost AGV for Distributed Production Lines and Warehouses. Em *CONTROLO 2008: 8th Portuguese Conference on Automatic Control*, páginas 827–832. University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, Julho 2008.
- [16] Automated Guided Vehicle Brochure. Disponível em [http://www.ocme.it/website/get\\_download.aspx?ctrb\\_id=177](http://www.ocme.it/website/get_download.aspx?ctrb_id=177).
- [17] AGV Electronics. Disponível em <http://www.agve.se/>. Último acesso a 26 de Fevereiro de 2010.
- [18] Automatic Guided Vehicle (AGV) basics. Disponível em [http://www.mmh.com/article/356365-Automatic\\_guided\\_vehicle\\_AGV\\_basics.php#guiding](http://www.mmh.com/article/356365-Automatic_guided_vehicle_AGV_basics.php#guiding). Último acesso a 26 de Fevereiro de 2010.
- [19] Guidance Options for Automatic Guided Vehicles. Disponível em [http://www.jerviswebb.com/Products/guidance\\_options.aspx?pid=308&q=1\\_6\\_](http://www.jerviswebb.com/Products/guidance_options.aspx?pid=308&q=1_6_). Último acesso a 26 de Fevereiro de 2010.
- [20] Eric Royer, Jonathan Bom, Michel Dhome, Benoit Thuilot, Maxime Lhuillier, e François Marmouton. Outdoor Autonomous Navigation using Monocular Vision. Em *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, páginas 1253 – 1258, Agosto 2005.
- [21] Azhar Dilshad e Dr. Sunita Chaunhan. Latest Advancements in Mobile Robot Localization in Manufacturing Environment. 2010. School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore.
- [22] J. Gonzalez, J.L. Blanco, C. Galindo, A. Ortiz-de Galisteo, J.A. Fernandez-Madrigal, F.A. Moreno, e J.L. Martinez. Combination of UWB and GPS for indoor-outdoor vehicle localization. Em *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, páginas 1–6, Outubro 2007.
- [23] F. Duvallet e A.D. Tews. WiFi Position Estimation in Industrial Environments Using Gaussian Processes. Em *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, páginas 2216–2221, Setembro 2008.
- [24] Hongbo Wang, Ke Yu, e Bingyi Mao. Self-localization and obstacle avoidance for a mobile robot. *Neural Computing & Applications*, 18(5):495–506, 2009.
- [25] A. Ladd, Kostas E. Bekris, Algis P. Rudys, Dan S. Wallach, e Lydia E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20:555–559, 2004.
- [26] Ubisense - Precise Real-Time Location Systems and GIS Consulting. Disponível em <http://www.ubisense.net/>. Último acesso a 30 de Abril de 2010.

- [27] Guilherme N. DeSouza e Avinash C. Kak. Vision for Mobile Robot Navigation: A Survey. Em *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, páginas 237–267, Fevereiro 2002.
- [28] Jun Yu, Peihuang Lou, Xiaoming Qian, e Xing Wu. An Intelligent Real-time Monocular Vision-based AGV System for Accurate Lane Detecting. Em *CCCM '08: Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 2, páginas 28–33, Agosto 2008.
- [29] Rita Cucchiara, Emanuele Perini, e Giuliano Pistoni. Efficient Stereo Vision for Obstacle Detection and AGV Navigation. Em *ICIAP '07: Proceedings of the 14th International Conference on Image Analysis and Processing*, páginas 291–296. IEEE Computer Society, Setembro 2007.
- [30] M. Asif, M. R. Arshad, e P. A. Wilson. AGV Guidance System: An Application of Simple Active Contour for Visual Tracking. Em *Proceedings of World Academy of Science, Engineering and Technology*, volume 6, páginas 74–77, Junho 2005.
- [31] Jeehoon Park, YoungSu Park, e Sang Woo Kim. AGV Parking System using Artificial Visual Landmark. Em *ICCAS 2008: International Conference on Control, Automation and Systems*, páginas 1579 – 1582, Outubro 2008.
- [32] How Kiva Robots Help Zappos and Walgreens. Disponível em [http://www.businessweek.com/innovate/content/apr2009/id20090415\\_876420.htm](http://www.businessweek.com/innovate/content/apr2009/id20090415_876420.htm). Último acesso a 18 de Março de 2010.
- [33] R.V. Bostelman, T.H. Hong, e R. Madhavan. Towards AGV Safety and Navigation Advancement - Obstacle Detection using a TOF Range Camera. Em *ICAR 2005: International Conf. On Advanced Robotics, July 18-20, 2005*.
- [34] JBT Corporation - Safety. Disponível em <http://www.jbtc-agv.com/knowledge/safety.aspx>. Último acesso a 18 de Março de 2010.
- [35] AGV-1000 Ultrasonic Collision Avoidance System. Disponível em <http://www.apgsensors.com/ultrasonic-sensor/agv-1000.html>. Último acesso a 28 de Abril de 2010.
- [36] Lundahl Sensors and Controller Systems. Disponível em [http://www.omnicontrols.com/lundahl\\_sensors.aspx](http://www.omnicontrols.com/lundahl_sensors.aspx). Último acesso a 28 de Abril de 2010.
- [37] Eagle E210 Flexcart. Disponível em <http://www.coreconagvs.com/products/E210R.php>. Último acesso a 28 de Abril de 2010.
- [38] Maxon Motor. Disponível em <http://www.maxonmotor.com/>. Último acesso a 28 de Abril de 2010.
- [39] Ana Ferreira. Extracção de Dimensões de Objectos por Laser para Integração com Manipuladores Industriais. Trabalho Final de Licenciatura em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2007.
- [40] Scanning Laser Range Finder URG-04LX - Specifications. Disponível em <http://www.hokuyo-aut.jp/02sensor/07scanner/download/index.html>. Último acesso a 3 de Maio de 2010.

- [41] António Paulo Moreira. Sistemas Robóticos Autónomos, Janeiro 2010. 1 diaporama (107 diapositivos). Disponível na Faculdade de Engenharia da Universidade do Porto.
- [42] Johann Borenstein e Liqiang Feng. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics and Automation*, 12:869–880, 1996.
- [43] Frédéric Chenavier e James Crowley. Position Estimation for a Mobile Robot Using Vision and Odometry. Em *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, páginas 2588–2593, Maio 1992.
- [44] Heber Sobreira. Clever Robot. Tese de Mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2009.
- [45] Taeyeon Kim e Joon Lyou. Indoor Navigation of Skid Steering Mobile Robot using Ceiling Landmarks. Em *ISIE 2009: IEEE International Symposium on Industrial Electronics*, páginas 1743–1748, Julho 2009.
- [46] Sooyong Lee e Jae-Bok Song. Mobile Robot Localization using Infrared Light Reflecting Landmarks. Em *ICCAS '07: International Conference on Control, Automation and Systems, 2007*, páginas 674 –677, Outubro 2007.
- [47] Armida González Lorence, no Gaffare, Mayra P. Gardu e J. Armando Segovia de los Ríos. Mobile Robot Global Localization using just a Visual Landmark. Em *ISPRA'06: Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation*, páginas 98–103, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).
- [48] Tohru Katayama. *Subspace Methods for System Identification*. Springer-Verlag, 2005.
- [49] Greg Welch e Gary Bishop. An Introduction to the Kalman Filter. Relatório té, Chapel Hill, NC, USA, 1995.
- [50] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 8(6):679–698, November 1986.
- [51] Bill Green. Canny Edge Detection Tutorial, 2002. Disponível em [http://www.pages.drexel.edu/~weg22/can\\_tut.html](http://www.pages.drexel.edu/~weg22/can_tut.html). Último acesso a 19 de Março de 2010.
- [52] Richard O. Duda e Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [53] Jiang-Ping He e Yan Ma. Triangle Detection Based on Windowed Hough Transform. Em *ICWAPR 2009: International Conference on Wavelet Analysis and Pattern Recognition 2009*, páginas 95 –100, Julho 2009.
- [54] Marcos Ferreira. Pintura Robotizada com Adaptação Automática. Tese de Mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2009.
- [55] Eric Weisstein. Line-Line Intersection. Disponível em <http://mathworld.wolfram.com/Line-LineIntersection.html>. Último acesso a 1 de Abril de 2010.